

von Karman Institute for Fluid Dynamics
Chaussée de Waterloo, 72
B - 1640 Rhode Saint Genèse - Belgium

CONTRACT REPORT 2003-08

December 2002

LES INVESTIGATION OF COHERENT STRUCTURES
IN BOUNDARY LAYERS AND WAKES
VOLUME III: DEVELOPMENT OF A PARALLEL
UNSTRUCTURED GRID LES CODE

GRANT N00014-99-1-0834

G. Degrez & D. Snyder

DISTRIBUTION STATEMENT A

Approved for Public Release
Distribution Unlimited

20030320 110

EAR9944

AQ F03-06-1408

Abstract

The purpose of the present investigation is to assess the feasibility of simulating and studying coherent structures in turbulent shear layers, making use of Large Eddy Simulations (LES).

Volumes I and II were devoted to the investigation of coherent structures in LES solutions obtained using a structured-grid finite-difference code, first in an attached turbulent flow, and then in the wake behind a square cylinder.

The structured grid code used in Parts I & II, despite its multi-domain capability, is limited in the domain geometries it can handle. The third part of the work was therefore devoted to the development of an unstructured grid LES code able to compute turbulent flows over arbitrary two-dimensional geometries.

A combined finite-element/Fourier spectral space discretization scheme was selected, as it combines optimally the geometrical flexibility provided by the finite element scheme and the computational efficiency resulting from the decomposition in Fourier modes in the out-of-plane direction. Indeed, thanks to this decomposition, and a suitable treatment of the nonlinear convective terms, the 3D flow problem is transformed into a series of 2D problems in Fourier space, completely decoupled within each time step. In addition, the decoupling allows for an easy parallelization by partitioning the work in Fourier space rather than physical space. Also, the stabilized finite element technique selected for the in-plane discretization provides an accuracy superior to its finite volume counterpart on the same unstructured grid and for the same discretization stencil.

The development of the code was broken down in the following steps:

1. development of a two-dimensional unsteady laminar flow solver, and validation;
2. development of a three-dimensional unsteady laminar flow solver, and validation;
3. development of the three-dimensional LES code, validation, and application to an original flow problem.

Although originally planned, the extension of the coherent structures detection algorithm developed in Part I for the structured grid LES solver could not be carried out for lack of time, as the development and testing of the solver took more time than anticipated.

The computational results obtained with the present solver were found in excellent agreement with existing experimental and computational data for the test case of the flow over a circular cylinder. The original computation of the flow over a circular cylinder with splitter plate was also found to be in excellent agreement with experimental data.

These encouraging results suggest that the developed code has a great potential for further investigation of turbulent flows over general two-dimensional geometries. Nevertheless, improvements are possible and needed. Suggestions for such improvements, and further investigations are provided.

Acknowledgements

Present research has been funded by the Office Naval Research under Grant N00014-99-1-0834, under supervision of Program Officers Dr. Candace Wark and Dr. Patrick Purtell.

Table of Contents

Abstract	ii
Acknowledgements	v
List of tables	ix
List of figures	x
1. Introduction	1
1.1 Fundamental Concepts	1
1.1.1 Computational Fluid Dynamics	1
1.1.2 Meshes	2
1.1.3 The Finite-Element Method	3
1.1.4 Spectral Methods	6
1.1.5 Large-Eddy Simulations	6
1.2 Objectives	8
1.2.1 2D Solver (SFE2D)	10
1.2.2 3D Laminar Solver (SFE3D)	11
1.2.3 LES Solver (SFELES)	11
1.3 Related Works	12
1.4 Contributions	13
2. 2D Laminar Flow Solver	15
2.1 Galerkin Finite-Element Method	15
2.1.1 Continuum Problem	15
2.1.2 Weighted Residual Formulation	16
2.1.3 Galerkin Formulation	17
2.1.4 Finite Elements	18
2.2 Spatial Discretization	20
2.2.1 Galerkin Formulation	20
2.2.2 PSPG Formulation	27
2.3 Temporal Discretization	33
2.3.1 Pressure and Diffusion Terms	34
2.3.2 Convective Terms	35
2.3.3 SUPG/PSPG Formulation of the NS Equations	35
2.4 Linear System Solution	37
2.5 SFE2D Summary	38
3. 2D Laminar Flow Test Cases	42
3.1 Lid-Driven Flow in a Square Cavity	42
3.1.1 Problem Description	42
3.1.2 Computational Mesh	43

3.1.3 $\text{Re} = 100$	44
3.1.4 $\text{Re} = 1000$	45
3.1.5 $\text{Re} = 5000$	45
3.2 Backward-Facing Step Flow	55
3.2.1 Problem Description	55
3.2.2 Computational Mesh	56
3.2.3 Results	57
3.3 Circular Cylinder Flow	59
3.3.1 Problem Description	59
3.3.2 Computational Mesh	61
3.3.3 Results	62
4. 3D Laminar Flow Solver	66
4.1 Governing Equations	66
4.2 Temporal Discretization	67
4.3 Spectral Methods	68
4.4 Finite-Element/Spectral Discretization	68
4.4.1 Nodal Matrix Equation	72
4.4.2 Analytical Coefficient Evaluation	74
4.4.3 Fourier Symmetry	75
4.5 Treatment of the Convective Terms	75
4.5.1 Temporal Treatment	75
4.5.2 Pseudo-Spectral Approach	76
4.5.3 Nodal Convective Load Vector	79
4.5.4 Analytical Coefficient Evaluation	79
4.6 Parallelization	80
4.6.1 Parallelization Theory	80
4.6.2 Parallelization Implementation	82
4.6.3 Parallelization Performance	88
4.7 SFE3D Summary	88
5. 3D Laminar Flow Test Case	94
5.1 Circular Cylinder Problem Description	94
5.2 Computational Mesh	95
5.3 Results	97
6. Large-Eddy Simulation Solver	104
6.1 LES Approach to Turbulence	104
6.2 Governing Equations	108
6.3 Subgrid-Scale Models	110
6.3.1 Smagorinski Model	110
6.3.2 Other SGS Models	112

6.4	Eddy Viscosity Implementation	114
6.4.1	Nodal Matrix Equation	114
6.4.2	Analytical Coefficient Evaluation	118
6.4.3	Smagorinski Model	119
6.5	Near-Wall Treatment	121
6.6	Aliasing Errors	122
7.	Large-Eddy Simulation Test Cases	126
7.1	Circular Cylinder	126
7.1.1	Problem Description	126
7.1.2	Computational Mesh	126
7.1.3	Results	129
7.2	Circular Cylinder With Wake Splitter Plate	139
7.2.1	Problem Description	139
7.2.2	Computational Mesh	141
7.2.3	Results	142
8.	Conclusion	148
8.1	Development of Navier-Stokes Solvers	148
8.1.1	SFE2D	148
8.1.2	SFE3D	148
8.1.3	SFELES	149
8.2	Turbulent Circular Cylinder Flow with a Wake Splitter Plate	149
8.3	Future Work	150
	APPENDIX	163
A.	Compressed Sparse Row (CSR) Matrix Format	164
B.	Input Data File Format (2D)	166
C.	Input Data File Format (3D)	169
D.	Commutative Properties of the LES Filter Operation	173

List of Tables

3.1	Flow over a backward facing step: comparison of detachment/reattachment lengths.	58
3.2	2D circular cylinder flow: comparison of results from commercial CFD solvers for drag coefficient (C_D) and Strouhal Number (St) at $Re = 60$	65
5.1	3D circular cylinder flow: comparison of C_D and St for SFE3D, published experiments, and published numerical results at $Re = 300$	100
7.1	Turbulent circular cylinder flow: comparison of flow parameters from SFELES at $Re = 3900$ with various experiments.	132
7.2	Turbulent circular cylinder splitter plate flow: mesh statistics summary.	141
7.3	Turbulent circular cylinder splitter plate flow: variation of C_{D_P} , St , and L_f with L/D	144

List of Figures

1.1 Comparison of 2D meshes: (a) structured quadrilateral, (b) structured triangle, (c) unstructured quadrilateral, (d) unstructured triangle, (e) semi-structured (multi-block structured) quadrilateral.	3
1.2 Shape functions for the P1 (linear triangle) element corresponding to (a) node i , (b) node j , and (c) node k	4
1.3 The P2/P1 element: a triangle element using quadratic interpolation for the velocity DOFs and linear interpolation for the pressure DOFs.	6
1.4 A possible future application of the 3D finite-element/spectral LES algorithm is that of high-lift multi-element airfoils.	10
2.1 One-dimensional elements.	18
2.2 One-dimensional, piecewise linear basis function N_j belonging to node j ,	19
2.3 Two-dimensional, tent-shaped, piecewise linear basis function N_j belonging to node j of a mesh of P1 elements.	19
2.4 The P1/P1 element: a triangle element using the same linear interpolation functions for the velocity and pressure DOFs.	21
2.5 Definition of the scaled inward normal vectors for a linear triangle element.	25
2.6 A hypothetical ‘checker-board’ pressure field on a uniform finite-difference mesh.	26
2.7 The lid-driven cavity test case: (a) relevant geometry and boundary conditions, and (b) unstructured triangle mesh containing approximately 3,000 nodes and 5,750 elements.	27
2.8 Stokes flow lid-driven cavity Galerkin results: (a) calculated velocity vectors, and (b) calculated pressure field corrupted by spurious oscillations.	28
2.9 Stokes flow lid-driven cavity PSPG results: (a) calculated velocity vectors, and (b) calculated pressure field (25 contours over the range $-20 \leq p \leq 20$).	30
2.10 Velocity vectors showing convection-induced oscillations in the PSPG solution of flow past a square block at $Re = 200$	31
2.11 Comparison of the weighting functions for the Galerkin and SUPG methods.	33
2.12 Velocity vectors showing the oscillation-free SUPG/PSPG coarse-mesh velocity solution of flow past a square block at $Re = 200$	33
2.13 Velocity vectors showing the oscillation-free PSPG fine-mesh velocity solution of flow past a square block at $Re = 200$	34
2.14 Flow chart showing the basic structure of the SFE2D Navier-Stokes solver.	41
3.1 Relevant geometry and boundary conditions for the lid-driven cavity test case.	42
3.2 Unstructured triangle mesh utilized for the lid-driven cavity test case, containing 3013 nodes and 5752 elements.	44
3.3 Triangle elements: (a) isotropic triangle, (b) high aspect ratio triangle poorly suited for numerical computations, (c) high aspect ratio ‘wedge’ well suited for numerical computations.	44

3.4	Mesh in the lower-left corner of the domain, $0 \leq x, y \leq 0.05$, showing the transition from wedge-shaped triangles to isotropic triangles in the corners. . .	45
3.5	Lid-driven cavity flow ($Re = 100$): streamtrace visualization of the flow field. .	46
3.6	Lid-driven cavity flow ($Re = 100$): velocity vector visualization of the flow field.	47
3.7	Lid-driven cavity flow ($Re = 100$): contour plot visualization of the pressure field.	47
3.8	Lid-driven cavity flow ($Re = 100$): comparison of the results of this work with the benchmark solution of Ghia et al (1988).	48
3.9	Lid-driven cavity flow ($Re = 1000$): streamtrace visualization of the flow field.	49
3.10	Lid-driven cavity flow ($Re = 1000$): velocity vector visualization of the flow field.	49
3.11	Lid-driven cavity flow ($Re = 1000$): contour plot visualization of the pressure field.	50
3.12	Lid-driven cavity flow ($Re = 1000$): comparison of the results of this work with the benchmark solution of Ghia et al (1988).	51
3.13	Lid-driven cavity flow ($Re = 5000$): streamtrace visualization of the flow field.	52
3.14	Lid-driven cavity flow ($Re = 5000$): velocity vector visualization of the flow field.	52
3.15	Lid-driven cavity flow ($Re = 5000$): contour plot visualization of the pressure field.	53
3.16	Lid-driven cavity flow ($Re = 5000$): comparison of the results of this work with the benchmark solution of Ghia et al (1988).	54
3.17	Flow over a backward-facing step: geometry with pertinent lengths and general flowfield properties.	56
3.18	Flow over a backward-facing step: computational mesh in the region of the step.	57
3.19	Flow over a backward-facing step: contours of constant streamwise velocity in the region $-1 \leq x/S \leq 27$ for Reynolds number (a) 150, (b) 500, and (c) 1000.	57
3.20	Flow over a backward-facing step: contours of constant pressure in the region $-3 \leq x/S \leq 8$ for Reynolds number 500.	58
3.21	Flow over a backward-facing step: streamwise velocity profiles at three locations in the channel and $Re = 1095$	59
3.22	2D circular cylinder flow: experimental flow visualizations from VanDyke (1982).	60
3.23	2D circular cylinder flow: overview of the computational domain and mesh. . .	62
3.24	2D circular cylinder flow: the computational mesh near the cylinder.	63
3.25	2D circular cylinder flow: 50 instantaneous contours of constant vorticity magnitude ($0 \leq \omega D/U_0 \leq 150$) in the region $-1.0D \leq x \leq 15.0D$, $-2.0D \leq y \leq 2.0D$. (a) $Re_D = 20$, (b) $Re_D = 60$, (c) $Re_D = 100$, (d) $Re_D = 140$	64
3.26	2D circular cylinder flow: quantitative comparison with published values of C_{D_p} , C_{D_v} , and St vs. Reynolds number.	65

4.1	Illustration of the symmetry properties of the Fourier coefficients when the physical-space values are purely real.	76
4.2	Illustration of the pseudo-spectral approach utilized in evaluation of the non-linear terms.	77
4.3	Approaches to partitioning for parallelization: (a) partitioning in physical space, (b) partitioning in Fourier space.	81
4.4	A simplified model of the shared-memory computer architecture, where each of the n processors has direct access to all system memory.	83
4.5	A simplified model of the distributed-memory computer architecture, where each of the n processors has direct access only to its own memory and access to memory located on another processor requires passing information through the interconnecting network.	84
4.6	Static load scheduling: assignment of work chunks to processors is fixed and determined prior to execution.	85
4.7	Dynamic load scheduling: assignment of work chunks to processors occurs dynamically during execution on a first-completed-first-served basis.	86
4.8	Fork-join parallelism model: the master thread spawns a team of threads inside a parallel region.	86
4.9	Application speedup as a function of the number of processors used.	89
4.10	Flow chart showing the basic structure of the SFE3D Navier-Stokes solver. . .	93
5.1	3D circular cylinder flow: mode-A and mode-B instabilities in the cylinder wake as visualized by Williamson (1996).	95
5.2	3D circular cylinder flow: overview of the computational domain and mesh containing 33,000 in-plane triangle elements (17,000 in-plane nodes) and 32 Fourier modes.	96
5.3	3D circular cylinder flow: surfaces of constant transverse vorticity showing the 2D Karman vortex street at $Re = 100$ after the transverse perturbation imposed at the inlet has washed downstream.	97
5.4	3D circular cylinder flow: mode-A instabilities in the wake at $Re = 195$	98
5.5	3D circular cylinder flow: mode-A instabilities in the wake, as calculated by Thompson et al (1994).	99
5.6	3D circular cylinder flow: mode-B instabilities in the wake at $Re = 300$	100
5.7	3D circular cylinder flow: mode-B instabilities in the wake, as calculated by Thompson et al (1994).	101
5.8	3D circular cylinder flow: mean streamwise velocity at different streamwise locations in the wake at $Re = 300$	101
5.9	3D circular cylinder flow: mean lateral velocity at different streamwise locations in the wake at $Re = 300$	102
5.10	3D circular cylinder flow: time-averaged streamwise velocity fluctuations at different streamwise locations in the wake at $Re = 300$	102

5.11	3D circular cylinder flow: time-averaged lateral velocity fluctuations at different streamwise locations in the wake at $Re = 300$	103
5.12	3D circular cylinder flow: time-averaged Reynolds shear stress at different streamwise locations in the wake at $Re = 300$	103
6.1	Comparison of the different approaches to handling turbulence. a) Direct Numerical Simulation (DNS), b) Reynolds Averaged Navier-Stokes (RANS), c) Large Eddy Simulation (LES).	106
6.2	Commonly-used filter functions. a) tophat, b) Gaussian, and c) sharp Fourier cutoff.	107
6.3	Filtering of a 1D function. a) tophat filter function, b) Gaussian filter function, c) sharp Fourier cutoff filter function, and d) sharp Fourier cutoff filter function with the filter width reduced by a factor of eight.	124
6.4	Attenuation factors of the three most common filter functions.	125
7.1	Turbulent circular cylinder flow: overview of the computational domain and mesh containing 23,500 in-plane triangle elements (12,000 in-plane nodes) and 32 Fourier modes.	127
7.2	Turbulent circular cylinder flow: view of the 2D mesh in the near-wake region.	128
7.3	Turbulent circular cylinder flow: view of the 2D mesh very near the cylinder, showing the wedge-type boundary layer elements for the (a) coarse and (b) fine meshes.	129
7.4	Turbulent circular cylinder flow: perspective view of instantaneous surfaces of constant vorticity magnitude, $\omega D/U_\infty = 2.5$	130
7.5	Turbulent circular cylinder flow: front view of instantaneous surfaces of constant vorticity magnitude, $\omega D/U_\infty = 2.5$, in the region $-0.5D \leq x \leq 10D$	131
7.6	Turbulent circular cylinder flow: bottom view of instantaneous surfaces of constant vorticity magnitude, $\omega D/U_\infty = 2.5$, in the region $-0.5D \leq x \leq 10D$	132
7.7	Turbulent circular cylinder flow: instantaneous streamwise velocity contours on the $y = 0$ plane and in the region $-0.5D \leq x \leq 10D$	132
7.8	Turbulent circular cylinder flow: instantaneous lateral velocity contours on the $y = 0$ plane and in the region $-0.5D \leq x \leq 10D$	133
7.9	Turbulent circular cylinder flow: instantaneous transverse velocity contours on the $y = 0$ plane and in the region $-0.5D \leq x \leq 10D$	133
7.10	Turbulent circular cylinder flow: instantaneous surfaces of constant vorticity magnitude, $\omega D/U_\infty = 2.5$, in the very near wake at five instants during the shedding cycle.	135
7.11	Turbulent circular cylinder flow: mean streamwise velocity along the line $y = 0$ at $Re = 3900$	136
7.12	Turbulent circular cylinder flow: mean streamwise velocity at different streamwise locations in the wake at $Re = 3900$	136

7.13	Turbulent circular cylinder flow: mean lateral velocity at different streamwise locations in the wake at $Re = 3900$	137
7.14	Turbulent circular cylinder flow: time-averaged streamwise velocity fluctuations at different streamwise locations in the wake at $Re = 3900$	137
7.15	Turbulent circular cylinder flow: time-averaged lateral velocity fluctuations at different streamwise locations in the wake at $Re = 3900$	138
7.16	Turbulent circular cylinder flow: time-averaged Reynolds shear stress at different streamwise locations in the wake at $Re = 3900$	138
7.17	Summary of the effects of a splitter plate on the formation region behind a circular cylinder.	140
7.18	Turbulent circular cylinder splitter plate flow: in-plane mesh near the cylinder for (a) $L/D = 0.25$, (b) $L/D = 0.75$, and (c) $L/D = 1.50$	142
7.19	Turbulent circular cylinder splitter plate flow: instantaneous surfaces of constant vorticity magnitude, $\omega D/U_\infty = 2.5$. Surfaces are colored according to velocity magnitude, with blue representing low values and red representing high values.	145
7.20	Turbulent circular cylinder splitter plate flow: effect of splitter plate length on the formation region behind the cylinder. Experimental flow visualizations compared with predicted surfaces of constant vorticity magnitude.	146
7.21	Turbulent circular cylinder splitter plate flow: pressure distribution around the cylinder for various splitter plate lengths.	147
B.1	Mesh corresponding to the sample DPlot data file, where <u>e</u> denotes an element and <u>n</u> denotes a node.	166
C.1	Mesh corresponding to the sample modified DPlot data file, where <u>e</u> denotes an element and <u>n</u> denotes a node.	169

1. INTRODUCTION

The work presented in this dissertation contributes to the numerical simulation of unsteady, turbulent, incompressible flows via the development of a large-eddy simulation (LES) solver for complex two-dimensional (2D) geometries using a combined finite-element/spectral method. This mixed spatial discretization coupled with certain time discretizations reduces the computational cost, both in terms of operation counts and storage requirements.

This introductory chapter first presents some fundamental concepts related to the development of the solution algorithm. A discussion of the objectives and contributions of the work as well as a brief presentation of some related works follows.

1.1 Fundamental Concepts

1.1.1 Computational Fluid Dynamics

Computational fluid dynamics (CFD) is the numerical simulation of fluid-flow systems. Initially restricted to coarse approximations of simple problems, CFD has developed to the stage where it now complements experiments in complex research and design applications. The two factors driving CFD forward are the increasing performance and decreasing cost of computer hardware, and the increasing efficiency of numerical algorithms.

From basic conservation principles, the partial differential equations (PDEs) governing fluid flow can be derived. For unsteady, isothermal, incompressible flow of a Newtonian fluid, these governing equations are the Navier-Stokes (NS) equations (conservation of momentum) and the continuity equation (conservation of mass). The equations are formulated in terms of the primitive variables, that is velocity \vec{u} and pressure p . In the absence of body forces, the incompressible NS equations become

$$\frac{\partial \vec{u}}{\partial t} + (\vec{u} \cdot \nabla) \vec{u} = -\nabla p + \nu (\nabla^2 \vec{u}) \quad (1.1)$$

and the continuity equation is

$$\nabla \cdot \vec{u} = 0. \quad (1.2)$$

In the above equations, t is time, ν the kinematic viscosity, p the kinematic pressure (pressure divided by density), and ∇ the gradient operator.

Unfortunately, because of the complexity of the equations and the geometry of the domains of interest, it is generally impossible to obtain analytical or closed-form solutions for the flow variables (velocity, pressure, etc.). The object of CFD, therefore, is the development of algorithms to compute an approximate solution to the mathematical problem.

This approximate solution is characterized by a finite number of unknowns, typically the flow variables at a set of points (nodes) within the domain of interest. Consequently, the computational domain is typically divided into non-overlapping, finite-dimensional cells or elements, with the flow variables calculated at characteristic points of the cells or elements, such as the vertices or centers of gravity.

For each unknown, an algebraic equation is derived from the differential problem and the boundary conditions governing the flow. The way by which this set of equations is derived is termed a discretization method. A number of discretization methods have been developed, the most common being the finite-difference method (FDM), the finite-volume method (FVM), and the finite-element method (FEM). These methods are discussed in numerous texts, including (Fletcher et al., 1991; Hirsch, 1988; Dick, 1993b,a). A less common, but nonetheless valuable discretization is the spectral method (Zienkiewicz, 1975).

1.1.2 Meshes

Figure 1.1 shows some typical space discretizations (meshes) for a simple 2D circular domain, each using approximately 175 nodes. The difference between structured, unstructured, and semi-structured meshes is easily observed.

Structured meshes, Figure 1.1(a,b), possess inherent ordered connectivity that allow for efficient solution techniques in the FDM and FVM. That is, all mesh points lie on the intersections of two families of curves, say a vertical mesh line i and a horizontal mesh line j . Consequently, each grid point can be uniquely identified via a set of integers as (i, j) . In terms of a computer algorithm, all nodes in the mesh can be visited by a simple loop over i and j . Neighboring nodes are easily identified as well, i.e. $(i+1, j)$, $(i-1, j)$, $(i, j+1)$, and $(i, j-1)$. However, structured meshes are very limited in the geometries that can be represented well. This can be seen from the simple circle example, where highly skewed elements appear.

The most significant advantage of unstructured meshes, Figure 1.1(c,d), is the flexibility to handle complex geometries. Additionally, unstructured meshes lend themselves very well to solution-adaptive mesh refinement/coarsening techniques. As expected, these advantages come at a price—unstructured meshes impose additional storage overhead because the element connectivity must be explicitly stored, and solution algorithms are typically more computationally expensive.

Semi-structured (sometimes termed multi-block structured) meshes, Figure 1.1(e) provide a compromise between structured and unstructured meshes. These meshes maintain some inherent ordered connectivity by combining “blocks” of structured nodes together in an unstructured arrangement. This type of mesh provides an alternative to unstructured meshes, but is less flexible and usually requires more user input to create. In addition, solvers based on semi-structured meshes tend to have accuracy troubles at the block interfaces.

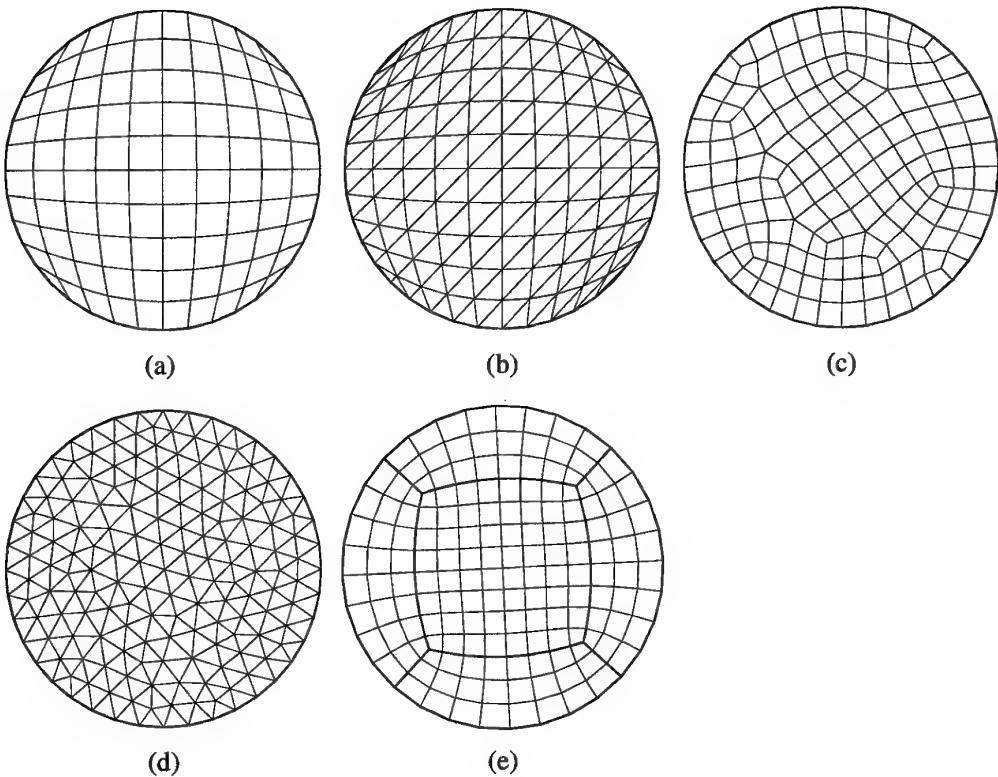


Fig. 1.1 – Comparison of 2D meshes: (a) structured quadrilateral, (b) structured triangle, (c) unstructured quadrilateral, (d) unstructured triangle, (e) semi-structured (multi-block structured) quadrilateral.

Finally, it should be stated that though structured meshes can, of course, be used with the FEM, the method is inherently realizable on unstructured meshes.

1.1.3 The Finite-Element Method

The FEM, just like the FDM and FVM, is simply a method for approximately solving PDEs. Upon being introduced to the FEM, one immediately notices the somewhat abstract mathematical character of the method in contrast to the physical-type nature of the FVM. This is, in fact, an advantage to the method in that applied mathematicians have provided a solid mathematical foundation. And, as asserted by Gresho (Gresho and Sani, 2000), “most of the difficult mathematics can be bypassed by practitioners and even code builders.” There are some definite advantages to the FEM that will be highlighted in this brief introduction to the method.

The first characteristic of the method is that the domain is subdivided into cells, called

elements, which form a mesh. There is no restriction as to the geometric form of the elements, though triangle or quadrilateral forms are typically seen in two dimensions (tetrahedron and hexahedron are most common in three dimensions). The first and most important advantage of the FEM is that this mesh can be unstructured, allowing complex geometries to be handled with relative ease. The FVM, but not traditional FDMs share this feature.

The elements consist not only of nodes and edges, but include shape functions (one for each node) that define how the solution varies inside the element. These shape functions are also commonly termed interpolation functions or trial functions. This defines a function space in which the solution to the PDE must belong. In the case of linear triangle elements (so-called P1 elements), which are used in this work, one considers a continuous function that varies linearly over each element, as shown in Figure 1.2.

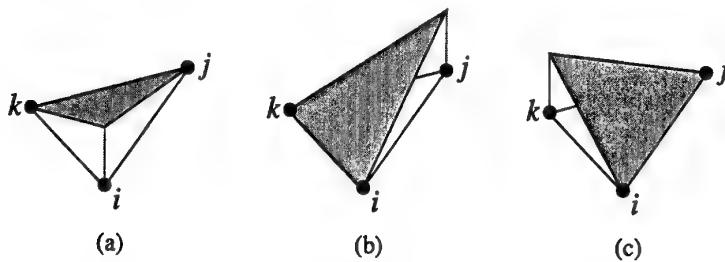


Fig. 1.2 – Shape functions for the P1 (linear triangle) element corresponding to (a) node i , (b) node j , and (c) node k .

The final characteristic of the FEM that will be discussed here is that the method does not attempt to find an approximate solution to the PDE itself, but rather for a solution of some integral form of the PDE. This integral form is most commonly—but not necessarily—obtained from a weighted residual formulation. This entails multiplying the PDE with predefined weight functions and integrating over the domain. The Galerkin formulation is obtained if the form of the shape functions is also employed for the weight functions. If the form of the shape and weight functions differ, a Petrov-Galerkin formulation is obtained. This weighted residual approach gives rise to the second major advantage to the FEM—the ability to naturally handle Neumann boundary conditions. Neither the FDM nor FVM share this feature.

When the Galerkin finite-element (FE) formulation is applied to the Navier-Stokes equations, two stability problems are encountered. First, at high Reynolds numbers (convection-dominated flow), oscillations appear in the approximate velocity solution. This phenomenon, which is also seen with the central difference counterparts in the FDM and FVM, can be handled in two ways.

- Imposing a restriction on the element Peclet number: $P_e \equiv \|\vec{u}\|h/\nu < 2$, where h is the characteristic element length in the direction of the local velocity \vec{u} .

- Introducing upwinding.

The former, restricting $P_e < 2$, guarantees convective stability but at the expense of high numbers of elements. That is, as the Reynolds number of the flow is increased, the mesh must be severely refined—usually much more so than is required for adequate resolution of the general flowfield. Because of this drawback, a number of upwinding schemes have been developed in the FE framework, the most common being the Streamline-Upwind Petrov-Galerkin (SUPG) formulation. This formulation is advantageous because it maintains the second-order accuracy of the Galerkin formulation and is also not subject to the artificial cross-flow diffusion criticisms associated with many classical upwind methods.

In addition to the above-mentioned convection-induced oscillations, Galerkin discretizations (as well as collocated variable discretizations in the FDM and FVM) suffer from spurious pressure oscillations due to the so-called odd-even decoupling phenomenon. In the FE framework, these instabilities are formulated as a violation of the so-called Babuška-Brezzi (Babuska, 1971, 1973; Brezzi, 1974) condition. The practical restriction placed by the Babuška-Brezzi condition is that the pressure interpolation must be of lower order than the velocity interpolation. These spurious pressure oscillations can be eliminated in one of two ways.

- Using a compatible element (equivalent to using a staggered-grid approach in the FDM and FVM) that satisfies the Babuška-Brezzi condition.
- Adding a small stabilizing term in the continuity equation that essentially filters out the spurious pressure oscillations (similar to Rhie-Chow in the FVM), bypassing the Babuška-Brezzi condition.

Until recently, the compatible Galerkin formulation has been the method of choice to eliminate the spurious pressure oscillations. Unfortunately, this technique has some implementation drawbacks, including considerably more complex data structures. A typical compatible element, the P2/P1 element, is shown in Figure 1.3. In this triangle element, the velocity interpolation is quadratic, while the pressure interpolation is linear.

To avoid the implementation drawbacks of compatible elements, the alternative pressure-stabilization approach has gained popularity in recent years. This latter approach is far more convenient from an implementation standpoint. As with all stabilization techniques, though, it is often difficult to determine the precise amount of stabilization that will eliminate the spurious oscillations but not spoil the accuracy of the solution.

In this work a Pressure-Stabilized Petrov/Galerkin (PSPG) formulation is used with P1/P1 (equal-order linear triangle) elements. This formulation is a full Petrov-Galerkin formulation in which a perturbed weight function is used to weight the continuity equation.

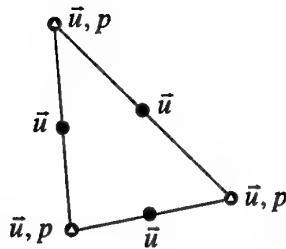


Fig. 1.3 – The P2/P1 element: a triangle element using quadratic interpolation for the velocity DOFs and linear interpolation for the pressure DOFs.

The Babuška-Brezzi condition is circumvented and equal-order interpolations are allowed. Not only can the PSPG method be interpreted as a Petrov-Galerkin formulation, but also as a generalized Galerkin method in which a stabilization term containing the residual of the momentum equation is added to the usual formulation. This formulation is consistent because as the solution converges, the stabilization term tends to zero.

For general information about the FEM given from a CFD point of view, the reader is referred to (Gresho and Sani, 2000). The interested reader is referred to Chapter 2 of this work, the thesis of DeMulder (DeMulder, 1997), or the article by Tezduyar (Tezduyar et al., 1992) for more detailed information on stabilized FE formulations.

1.1.4 Spectral Methods

Spectral methods approximate the unknown solution by means of a truncated Fourier series or a series of Chebyshev polynomials. The major difference between spectral discretizations and the FDM, FVM, or FEM is that the approximations are not local (compact), but are valid throughout the entire computational domain. As with the other methods, the unknowns in the governing equation are replaced by the truncated series approximation. The system of algebraic equations are then developed by means of a weighted residual approach (as discussed previously) or, alternatively, by forcing the approximate function to coincide with the exact solution at a number of grid points (termed a collocation approach).

Spectral methods have the advantage of being highly accurate for relatively few grid points. However, the drawbacks include increased computational cost and difficulty in applying boundary conditions for complex computational geometries. For more information on the use of spectral methods in CFD, the reader is referred to (Canuto et al., 1988; Hussaini and Zang, 1987).

1.1.5 Large-Eddy Simulations

Virtually all flows of engineering interest, whether in aerodynamics or industrial applications, are turbulent. Turbulence is a random eddying motion existing in high Reynolds number flows and containing a wide range of eddy sizes and fluctuation frequencies. The largest eddies have sizes on the same order of magnitude as the flow domain, low frequencies, and are highly affected by the flow boundaries. The smallest eddies, on the other hand, are on the order of the Kolmogorov length scale and have very high fluctuation frequencies. As the Reynolds number of the flow is increased, the difference between the largest and smallest eddies increases because the Kolmogorov length scale decreases.

Three principle techniques for handling turbulence in numerical approximations to the Navier-Stokes equations exist. Direct numerical simulation (DNS), which involves numerically solving the full unsteady Navier-Stokes equations, is currently limited to low Reynolds numbers and only the simplest flow geometries. This is because resolving all the length and time scales of turbulence requires computational resources that are prohibitively expensive (or often non-existent). This is illustrated by Emmons (Emmons, 1970), who shows that prediction of the fine details of turbulent flow in a pipe at Reynolds number $Re_D = 10^7$ would require 10^{22} operations. Assuming a ‘teraflop’ supercomputer (10^{-12} seconds per operation) was at your disposal and had sufficient memory, the computation would still take nearly 320 years!

Alternatively, the Reynolds-averaged Navier-Stokes (RANS) equations, obtained from time-averaging the unsteady Navier-Stokes equations, require much less in terms of computational resources and are used successfully to compute many flows of practical importance. Unfortunately, the turbulence models used in conjunction with the RANS equations are not applicable to a wide range of flow geometries and are inadequate for many turbulent flow situations. The major deficiency with the RANS equations is that all turbulence fluctuations are averaged, so the models must account for all scales of turbulence. Researchers have been developing turbulence models since the early 1970’s, but to date it has proved impossible to develop a universal turbulence model applicable to all flow situations.

LES is a compromise between DNS and RANS. LES relies on the fact that the small turbulent scales are nearly isotropic and independent of the geometry, whereas the large turbulent scales are mostly anisotropic and vary from flow to flow. The small scale motion, which is primarily an energy dissipation phenomenon and therefore more universal, is filtered out of the governing equations and modeled with a sub-grid scale (SGS) model. The large-scale motion, on the other hand, is computed directly by numerically solving the three-dimensional (3D), unsteady, filtered Navier-Stokes equations.

The scales of wavelength smaller than the mesh spacing Δx are removed by applying a low-pass filter function \overline{G} to the Navier-Stokes equations. The filtered variable corresponding

to some quantity f , denoted as \bar{f} , is defined as

$$\bar{f}(\vec{x}, t) = \int f(\vec{r}, t) \overline{G}(\vec{x} - \vec{r}) d\vec{r}. \quad (1.3)$$

After applying the above filtering to the Navier-Stokes and continuity equations, one arrives at the following set of governing equations, written in tensor notation

$$\frac{\partial \bar{u}_i}{\partial t} + \frac{\partial}{\partial x_j} (\bar{u}_i \bar{u}_j) = -\frac{\partial \bar{p}}{\partial x_i} + \frac{\partial \tau_{ij}}{\partial x_j} + \nu \frac{\partial}{\partial x_j} \left(\frac{\partial \bar{u}_i}{\partial x_j} \right) \quad (1.4)$$

$$\frac{\partial \bar{u}_i}{\partial x_i} = 0, \quad (1.5)$$

where the subgrid-scale Reynolds stress tensor τ_{ij} is given by

$$\tau_{ij} = \bar{u}_i \bar{u}_j - \bar{u}_i \bar{u}_j. \quad (1.6)$$

The main difficulty in LES is how to model this τ_{ij} term. Most SGS models make a Boussinesq approximation, in which the SGS Reynolds stress is assumed proportional to the super-grid, or resolved, strain. The resulting models are termed eddy-viscosity models.

Smagorinsky's model, proposed in 1963 (Smagorinski, 1963), is by far the most widely used. In this model, a mixing-length assumption is made in which the eddy viscosity is proportional to the characteristic length scale Δx and a characteristic turbulent velocity. This model has had problems in reproducing experimental results for some basic flows, including boundary layer transition (Piomelli, 1994) and flow past a backward-facing step (Lesieur and Metais, 1996). These problems are attributed to the model's overly-diffusive behavior, especially close to solid walls.

More recent SGS models have been developed, such as Kraichnan's spectral eddy viscosity model (Kraichnan, 1976), the structure-function (SF) model (Metais and Lesieur, 1992), and a class of models termed dynamic models. The strengths and weaknesses of each of these models will not be discussed here, but rather in Section 6.3.2. If interested in the current state of SGS models, the reader is referred to the excellent review article by Lesieur and Métais (Lesieur and Metais, 1996).

1.2 Objectives

Though less computationally intensive than DNS, LES calculations are nonetheless laborious and very much limited by current computer hardware. There are a number of characteristics of LES that add to this high computational cost, all of which arise from the necessity to calculate some turbulent scales. LES

- requires very fine spatial discretizations,

- is inherently unsteady and requires very fine temporal discretizations,
- and is inherently 3D.

Owing to the last characteristic listed above, there is no true ‘2D LES.’ However, one would hope that simplifications exist and can be exploited for LES computations utilizing 2D geometries.

If a 2D geometry and periodic flow is assumed in the transverse (z) direction, a spectral discretization is well-suited in that direction. In order to accommodate complex 2D geometries, however, a FE discretization is desired in the 2D plane. A combined FE/spectral discretization method is ideal. As will be illustrated later, this formulation results in several benefits, the most significant of which are listed below.

- Using an unstructured triangle mesh in the 2D plane allows easy accommodation of complex 2D geometries.
- Storage requirements are reduced because only a single 2D mesh need be stored.
- Computation and storage requirements are reduced because element areas and normal vectors need only be calculated and stored for a single 2D mesh.
- If an explicit treatment is used for the convective terms, the Fourier modes are decoupled within each time step and the problem is essentially converted to solving a number of independent 2D problems at each time step. This situation requires less computation than solving the 3D problem and also lends itself nicely to a parallel implementation.
- Because the flow variables in physical space are known to be real quantities, symmetry exists in the Fourier modes. Consequently, matrix equations for only $K/2 + 1$ Fourier modes must be solved (where K is the total number of modes) and the other modes can be quickly evaluated from the symmetry condition.

The main objective of this work is the development and implementation of a FE/spectral LES solver. By exploiting the advantages listed above, computational costs are reduced sufficiently to eventually allow LES calculations on very complex 2D geometries such as multi-element high-lift airfoils, as illustrated in Figure 1.4.

Three sequential tasks were defined in order to achieve the main objective.

- Development of a 2D FE incompressible Navier-Stokes solver (SFE2D).
- Extension of the 2D FE solver to a 3D combined FE/spectral solver (SFE3D).
- Addition of a subgrid-scale model, yielding a 3D FE/spectral LES solver (SFELES).

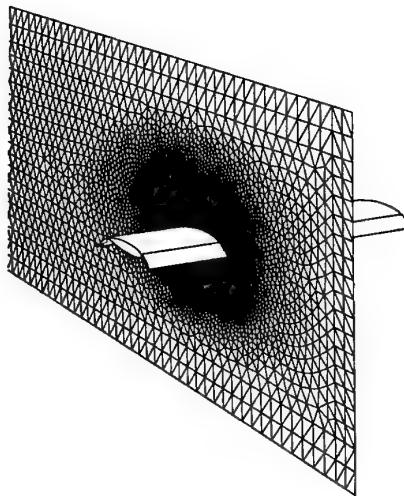


Fig. 1.4 – A possible future application of the 3D finite-element/spectral LES algorithm is that of high-lift multi-element airfoils.

Because this division of tasks had a profound effect on the development of the final algorithm, the structure of this dissertation also reflects this division.

1.2.1 2D Solver (SFE2D)

The first task was the development and implementation of a 2D FE incompressible Navier-Stokes solution algorithm. The solver, named SFE2D, was developed from the ground up and written in Fortran90.

The space discretization is formulated using a Galerkin FE discretization with P1/P1 elements. The pressure and convective instabilities associated with the Galerkin FEM are removed via a SUPG/PSPG formulation. This approach (SUPG/PSPG with P1/P1 elements) was chosen because of the relative ease of implementation resulting from a simple data structure as well as the ability to evaluate all coefficients in the matrix equation analytically.

This spatial discretization results in second-order spatial accuracy. The SFE2D solver is also second-order accurate in time. This is accomplished by means of a consistent mass matrix and Crank-Nicolson treatment of the pressure and diffusion terms. The convective terms are handled in an explicit manner—a second-order Adams-Bashforth method is utilized for these terms.

Solution of the algebraic system is performed using the SPARSKIT tool kit developed by Youcef Saad, University of Minnesota (Saad, 1994b). This library, written in Fortran77, was chosen because of its available compact matrix storage formats, efficient implementation

of a number of solution algorithms, and the inclusion of many preconditioning algorithms.

1.2.2 3D Laminar Solver (SFE3D)

The 3D laminar solver, referred to as SFE3D, was developed by extending the SFE2D solver. The in-plane and temporal discretization methods are identical to those in the SFE2D solver. The transverse (z) direction discretization uses a spectral method.

This solver takes advantage of the benefits listed previously. Grid overhead costs are reduced because the solver uses only a 2D mesh. The use of the explicit Adams-Bashforth method for the convective terms decouples the matrix equations for the Fourier modes, reducing the computation cost and allowing efficient parallelization. Finally, the Fourier transform symmetry characteristics are exploited to reduce the computations.

The algorithm is parallelized in three main areas. The first is in the transformation in the transverse direction between real space and Fourier space. Each set of nodes with the same in-plane coordinates is transformed independently of all other sets; therefore this transformation is carried out in parallel. The second area is in the solution of the independent linear systems for each Fourier mode within each time step. The third is in the evaluation of the convective terms, which is done on a plane-by-plane basis in physical space.

The major limitation imposed by parallelizing in this manner is that it must be implemented on a shared-memory computer. The reason being that for the parallelization of the linear system solutions, the domain is partitioned in the transverse direction. However, once the N independent solutions are obtained, information from all transverse planes is necessary to evaluate the convective terms. If this were to be implemented on a distributed memory system, the communication between processors would be prohibitively expensive.

The parallelization is implemented using the OpenMP standard. OpenMP is an emerging standard for shared-memory parallelism with considerable industry support from such large corporations as IBM, Intel, and Silicon Graphics. This standard allows for two basic ‘flavors’ of parallelization: segmentation of the code into different sections to be run as different threads, or execution of the iterations of a DO loop in parallel. The parallelization scheme implemented in SFES3D is a rather straightforward application of the DO loop flavor.

1.2.3 LES Solver (SFELES)

The LES solver, referred to as SFELES, is a direct extension of the SFE3D solver. The in-plane, transverse, and temporal discretizations are identical to those discussed in the previous sections.

The turbulence effects are modeled using the Smagorinski SGS model. These terms are evaluated using finite-elements in the 2D plane and finite-differences in the transverse

direction. Temporally, a second-order explicit Adams-Bashforth method is employed.

Wall functions are not implemented into the SFELES solver. Instead, the mesh must be sufficiently refined to resolve the near-wall gradients. Certainly, then, this solver cannot be used at very high Reynolds numbers because as the Reynolds number is increased, an increasing number of mesh points must be employed.

1.3 Related Works

The idea of employing different discretization schemes for different coordinate directions is not new. There are, in fact, numerous examples of mixed FD/spectral methods published in scientific literature. The motivating idea behind this approach is to use spectral methods in coordinate directions where periodicity exists, while using a FDM in coordinate directions where more complex boundary conditions must be applied.

Wray and Hussaini (Wray and Hussaini, 1984) used this technique to perform calculations of parallel boundary-layer transitions. They used a Fourier spectral method in two periodic directions (streamwise and transverse) and second-order finite-differences in the normal direction. A slightly different spectral/finite-difference method was used by Moin and Kim (Moin and Kim, 1982) for LES of turbulent channel flow and by Biringen (Biringen, 1985) in a study of active control in channel flows. Additionally, Eidsen, Hussaini, and Zang (Eidsen et al., 1986) used a similar algorithm in a high-resolution DNS study of a turbulent Rayleigh-Bénard flow.

Researchers in meteorology have been using so-called MSFD (Mixed Spectral Finite-Difference) algorithms for over a decade. Example studies include Beljaars et al (Beljaars et al., 1987), Ayotte et al (Ayotte et al., 1994), and Ayotte and Taylor (Ayotte and Taylor, 1995). In these cases, the horizontal coordinate directions are discretized using a spectral method, while the vertical coordinate is discretized using finite-differences.

The above-mentioned studies used Cartesian coordinate systems. Lu et al (Lu et al., 1997) employed a mixed FD/spectral method with cylindrical coordinates to study oscillating flows around a circular cylinder using LES. In this case, the circumferential and axial coordinate directions were discretized using a spectral method while a FDM was used in the radial direction. Yuan and Prosperetti (Yuan and Prosperetti, 1994) used a mixed FD/spectral method with a bispherical coordinate system to study the motion of two equal spherical bubbles moving along their line of centers.

LES solvers employing unstructured meshes, particularly the FEM, are rather uncommon. The reason being that the computational cost of LES is very high, and very fast algorithms typically employing Cartesian, cylindrical, or spherical meshes are preferred. Unstructured meshes allow the possibility of simulating flows with complex geometries, however, and are beginning to appear. Haworth and Jansen (Haworth and Jansen, 2000) developed an

unstructured LES algorithm based on the FVM. Their solver, which employed a number of variations of the Smagorinski SGS model, was used with some success in simulating flow in a piston/valve system using deforming meshes. Miura et al (Miura et al., 1999) employed a Petrov-Galerkin FEM with linear hexahedral elements and the Smagorinski model in solving turbulent flow past a square cylinder. Mittal (Mittal, 2001) developed an LES algorithm based on a SUPG/PSPG FE formulation with an additional least-squares convective stabilization. Using linear hexahedral elements, he successfully applied this solver to flow past a low aspect ratio cylinders in a channel with sidewalls. Finally, Jansen (Jansen, 1999) used a stabilized FE formulation with linear tetrahedral elements and a dynamic SGS model. The goal of his work was to simulate flow past a finite-span wing at $Re = 1.64 \times 10^6$, but only partial success was realized. The grid-refined simulations would have required on the order of 70 full days to complete using all 512 processors of the CM5 supercomputer at the Army High Performance Computing Research Center, so the study was not fully completed.

1.4 Contributions

The SFELES solver, produced as part of this work, is a LES solver for 2D geometries that utilizes a combined FE/spectral discretization. Specifically, the in-plane discretization is a stabilized FEM with linear triangle elements, while the transverse discretization uses a collocated pseudo-spectral approach. The unique characteristics of this solver are now summarized.

- Though stabilized FE solvers have been around for some time, to the author's knowledge, this is the first time a stabilized FEM has been coupled with a spectral method.
- SFELES is a LES solver designed to exploit the simplifications accompanying a 2D geometry. Although the flowfield is 3D, only a 2D mesh must be generated and stored. Similarly, the overhead associated with the mesh, including element areas, normals, and linkages, are only computed for the 2D mesh. Compared to a typical 3D FE solver, a decrease in computational cost is achieved because the 3D problem is converted into a set of independent 2D problems in Fourier space via careful treatment of the nonlinear terms.
- The use of unstructured meshes with LES is rather new due to the associated increase in computational cost. Exploiting the simplifications accompanying 2D geometries helps compensate for this cost, and the unstructured mesh allows for very complex geometries in the 2D plane.
- The solver is parallelized using a novel approach that partitions the problem in Fourier space rather than physical space. This results in no partitioning overhead in terms of computation cost, and no convergence or accuracy problems related to partitioning.

Also, there is absolutely no communication between threads during the solution of the linear systems generated at each time step.

In addition to SFELES, two intermediate solvers have also been developed. SFE2D is a 2D unsteady, laminar NS solver based on a stabilized FEM and using linear triangle elements. The SFE3D solver is a 3D unsteady, laminar NS solver based on a stabilized FE/spectral discretization.

The algorithm developed and implemented in this work has been applied to flow past a circular cylinder with a splitter plate in the wake region. This study has resulted in contributions to the understanding of the physical mechanisms involved in vortex formation in the presence of splitter plates as well as the behavior of shedding frequency and drag with varying splitter plate lengths.

2. 2D LAMINAR FLOW SOLVER

The purpose of this chapter is to explain the development of SFE2D, a 2D, unsteady, laminar NS solver utilizing a SUPG/PSPG FEM.

First, the spatial discretization is presented. The Galerkin formulation of the Stokes equations is presented in some detail and applied to the case of lid-driven cavity flow in order to illustrate its stability problems. To resolve these problems, the PSPG formulation of the Stokes equations is developed and applied to the same lid-driven cavity test case.

The addition of the convective terms to the Stokes equations results in the steady NS equations. These terms add a new instability to the FEM formulation, placing a restriction on the allowable cell Peclet number. To illustrate this stability problem, the PSPG algorithm is applied to flow past a rectangular bump in a channel. To remedy this stability issue, the SUPG/PSPG formulation of the steady NS equations is developed and applied to the same test case.

Next, the temporal discretization scheme is presented. The diffusion and pressure gradient terms in the momentum equations are discretized using a Crank-Nicolson scheme, while the convection terms use an explicit Adams-Bashforth scheme. The terms added to the linear system are derived and presented.

Finally, an overview of SFE2D is given, including some detail regarding implementation.

2.1 Galerkin Finite-Element Method

A presentation of the basic components of a Galerkin FEM is given in this section. The complete process to obtain a set of linear equations for the unknowns at the nodes is not shown here, but rather saved until the next section where it is demonstrated for the Stokes equations.

2.1.1 Continuum Problem

Consider a 1D linear boundary value problem (BVP) consisting of the differential equation

$$L(u(x)) = f(x) \quad \text{for } x \in \Omega \tag{2.1}$$

and boundary conditions

$$\begin{aligned} u &= \bar{u} && \text{on } \Gamma_{du} \\ \lambda \frac{\partial u}{\partial x} &= q && \text{on } \Gamma_{nu} \end{aligned} \tag{2.2}$$

where Ω is the domain on which (2.1) is valid and Γ is the boundary of Ω . Note that Γ is split into two parts: a part containing Dirichlet boundary conditions Γ_{du} and a part containing Neumann boundary conditions Γ_{nu} .

2.1.2 Weighted Residual Formulation

The first step in arriving at a Galerkin FEM solution is to recast the continuum problem (2.1) into its weak form, also referred to as its variational form. Though the continuum formulation of the BVP, (2.1, 2.2), is generally unique, there exists no unique weak formulation of the same problem. Though alternate weak formulations may exist for the BVP, they are actually equivalent as long as a solution to the BVP exists and is sufficiently smooth. As Gresho (Gresho and Sani, 2000) states, “Some weak formulations... are more useful than others because (at least) they more efficiently and more ‘naturally’ take account of the [boundary conditions].”

A finite-dimensional space H^h to which the approximate solution will belong is defined. Then, an approximate solution to the BVP (2.1, 2.2) having the form

$$u^h(x) = \sum_{j=1}^n u_j N_j(x) \quad (2.3)$$

is searched for, where u_j are the so-called degrees of freedom, the unknowns of the discrete problem. The basis functions N_j are finite-order polynomials that make up H^h . The dimension of the function space H^h is finite, $H^h = \{N_j : j = 1, 2, \dots, n\}$, therefore (2.3) cannot in general satisfy (2.1) at all points in the domain Ω . The basis functions, however, are chosen such that they form a complete set. That is, as n grows the approximation obtained by (2.3) improves.

Because u^h cannot satisfy (2.1) exactly, if (2.3) is substituted into (2.1) a residual remains:

$$r_\Omega = L(u^h) - f \quad \text{in } \Omega. \quad (2.4)$$

A natural approach to arriving at an approximate solution to the BVP is to ensure that this residual is somehow minimized. This can be done in a number of ways.

One approach would be to require that the square of the residual be a minimum over the entire domain Ω . That is, minimize

$$\int_\Omega (r_\Omega)^2 d\Omega. \quad (2.5)$$

This is known as a least squares approach.

Another approach common to the FEM is the method of weighted residuals. In this formulation, the residual multiplied by some weighting function is minimized over Ω . That

is, minimize

$$\int_{\Omega} w_i r_{\Omega} d\Omega. \quad (2.6)$$

The set of weighting functions $W = \{w_i : i = 1, 2, \dots, n\}$ must form a complete set and be the same in number as the basis functions N_j .

2.1.3 Galerkin Formulation

The Galerkin formulation is one of the many possible choices for the weighting functions. Though the Galerkin formulation is used in this work, two other choices for weighting functions are briefly mentioned to illustrate the generality of the FEM and its closeness to both the FDM and FVM.

In the collocation method, the weight functions are Dirac-delta functions in n points in the domain:

$$w_i = \delta(x - x_i); \quad i = 1, 2, \dots, n. \quad (2.7)$$

This essentially ensures that the residual be equal to zero at a number of points in the domain, and has much in common with the philosophy of the FDM.

Another choice of weighting functions, termed the subdomain collocation method, is to use step-discontinuous functions of the form

$$w_i = \begin{cases} 1, & \text{for } x_i \leq x \leq x_{i+1}, \\ 0, & \text{otherwise.} \end{cases} \quad (2.8)$$

This formulation ensures that the integral of the residual be zero on n subdomains, or elements. The FVM, which also utilizes an integral form of the continuum equation (rather than a differential form), is a particular case of this method.

Because the requirements for the weighting functions are that they form a complete set and be in number equal to the basis functions, an obvious choice is to use the basis functions themselves as the weighting functions:

$$w_i = N_i. \quad (2.9)$$

This formulation, which is the most common in the FEM, is termed the Galerkin method. This method has the advantage of simplicity in that only one set of functions is used. Also, using integration by parts on the differential operator allows the use of lower order basis functions. For example, second-order equations can be approximated using linear basis functions. Drawbacks include the need to evaluate possibly complex integrals and some stability issues for convection dominated flows.

2.1.4 Finite Elements

As will be shown, the FEM shares many successful concepts with the FDM. One is that the formulation is said to have compact support—the discretized equations couple only a few neighboring point values. In addition, the parameters of the approximate representation, the u_j 's in (2.3), are point values (as opposed to subdomain averages as in the FVM).

The computational domain Ω is divided into a set of non-overlapping subdomains Ω_e . In one dimension, an obvious choice for splitting up the domain would be as shown in Figure 2.1, where the subdomains extend between two adjacent points. In this figure two sample subdomains are highlighted: subdomain ${}^A\Omega_e$, the interval $x_{j-1} \leq x \leq x_j$, and subdomain ${}^B\Omega_e$, the interval $x_j \leq x \leq x_{j+1}$. In two dimensions the domain is divided into simple polygons (triangles, quadrilaterals, etc.) as depicted in Figure 1.1. Correspondingly, in three dimensions the domain is divided into simple polyhedra.

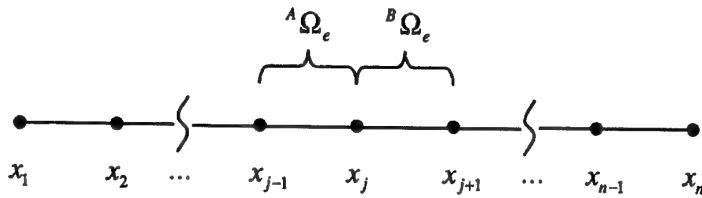


Fig. 2.1 – One-dimensional elements.

A set of nodes is associated to each subdomain, which typically includes its vertices. In addition, nodes may be located in the interior or along the subdomain faces. Together with its set of nodes, the subdomain is termed an element.

Each node i is assigned a basis function $N_i(x)$ in 1D or $N_i(x, y)$ in 2D, which is defined locally within the element. The basis functions are local interpolation functions within each element. Therefore, the basis function associated to node i must vanish at all other nodes of the element Ω_e . In other words, for all nodes j in Ω_e ,

$$N_i(x_j) = \delta_{ij} \quad (2.10)$$

in 1D, or similarly in 2D,

$$N_i((x, y)_j) = \delta_{ij} \quad (2.11)$$

where δ is the Kronecker delta. As a result of this property, a degree of freedom, u_j in (2.3), is the value of the approximate function u^h at node j , or

$$u_j = u^h(x_j). \quad (2.12)$$

Again, the basis functions are defined locally on each element. The global basis function associated to a node j is simply the combination of the basis function on each element to which node j belongs. The global basis function associated to node j is identically zero on all elements that do not contain the node, thus providing compact support.

In this work, linear basis functions are utilized. Therefore, higher order functions will not be discussed. Figure 2.2 shows 1D linear elements and the basis function associated with node j . It is seen that the global basis function N_j is the combination of the two local basis functions $^A N_j$ and $^B N_j$, defined on elements A and B respectively.

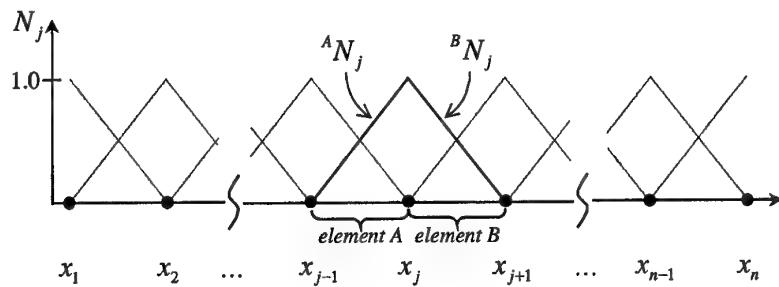


Fig. 2.2 – One-dimensional, piecewise linear basis function N_j belonging to node j .

This work utilizes 2D linear triangle elements. These elements are given the name P1, where P refers to the triangle shape and 1 to the order of the interpolation polynomials. The local basis functions associated with each of the three nodes of an element are depicted in Figure 1.2. The resulting tent-shaped, piecewise linear global basis function associated to a node j is illustrated in Figure 2.3.

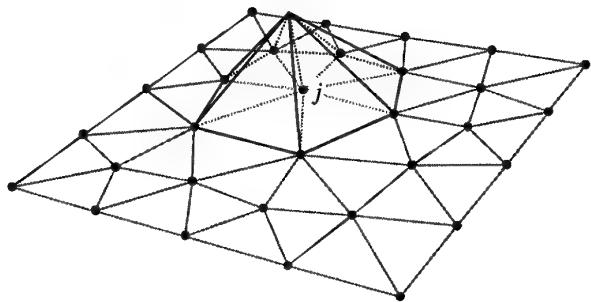


Fig. 2.3 – Two-dimensional, tent-shaped, piecewise linear basis function N_j belonging to node j of a mesh of P1 elements.

2.2 Spatial Discretization

The spatial discretization is best illustrated assuming steady flow. Since, as mentioned previously, the convection terms are treated explicitly, this section will focus on the discretization of the Stokes flow equations, also termed the creeping flow equations. These are the limit of the Navier-Stokes equations as $Re \rightarrow 0$.

Consider, then, creeping flow in the computational domain Ω with boundary $\Gamma = \Gamma_d \cup \Gamma_n$, where Γ_d is the portion of the boundary subject to Dirichlet boundary conditions and Γ_n the portion subject to Neumann conditions. The flow is governed by the momentum equation

$$\nu \nabla^2 \vec{u} - \nabla p = 0 \quad (2.13)$$

and the continuity equation

$$\nabla \cdot \vec{u} = 0. \quad (2.14)$$

In the above equations, ν is the kinematic viscosity of the fluid, \vec{u} is the 2D velocity vector (u, v) , and p is the kinematic pressure (pressure divided by density).

Equations (2.13) and (2.14) are subject to Dirichlet and Neumann boundary conditions:

$$\vec{u} = \bar{\vec{u}} \quad \text{on } \Gamma_d \quad (2.15)$$

$$(-p[I] + \nu \nabla \vec{u}) \cdot \vec{n} = \bar{\theta} \quad \text{on } \Gamma_n, \quad (2.16)$$

where $\bar{\vec{u}}$ is the imposed Dirichlet velocity boundary condition, $[I]$ is the identity matrix, \vec{n} is the outward unit normal on Γ , and $\bar{\theta}$ is the imposed surface traction boundary condition.

2.2.1 Galerkin Formulation

To obtain a FE formulation, the form of the approximate solution is first prescribed. We will denote the approximate solution for the velocity as (u^h, v^h) and the approximate solution for the pressure as p^h . Adopting the form of (2.3), we define

$$u^h = \sum_j u_j N_j(x, y) \quad (2.17)$$

$$v^h = \sum_j v_j N_j(x, y) \quad (2.18)$$

$$p^h = \sum_j p_j N_j(x, y) \quad (2.19)$$

where the N_j are the basis functions and u_j , v_j , and p_j are the unknown values of u , v , and p , respectively. In this work a P1/P1 element is utilized, meaning a linear triangle element with the DOFs for both the velocity and pressure located at the vertices of the element, as shown in Figure 2.4.

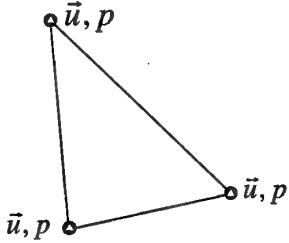


Fig. 2.4 – The P1/P1 element: a triangle element using the same linear interpolation functions for the velocity and pressure DOFs.

For purposes of illustration, (2.13) and (2.14) will be written in expanded form:

$$\nu \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right) - \frac{\partial p}{\partial x} = 0 \quad (2.20)$$

$$\nu \left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} \right) - \frac{\partial p}{\partial y} = 0 \quad (2.21)$$

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0. \quad (2.22)$$

A weighted residual formulation is obtained by weighting the momentum equation by ${}^{v_x}\bar{w}^h = ({}^{v_x}\bar{w}^h, {}^{v_y}\bar{w}^h)$ and the continuity equation by ${}^p w^h$, resulting in

$$\int_{\Omega} {}^{v_x} w_i \left[\nu \left(\frac{\partial^2 u^h}{\partial x^2} + \frac{\partial^2 u^h}{\partial y^2} \right) - \frac{\partial p^h}{\partial x} \right] d\Omega = 0 \quad (2.23)$$

$$\int_{\Omega} {}^{v_y} w_i \left[\nu \left(\frac{\partial^2 v^h}{\partial x^2} + \frac{\partial^2 v^h}{\partial y^2} \right) - \frac{\partial p^h}{\partial y} \right] d\Omega = 0 \quad (2.24)$$

$$\int_{\Omega} {}^p w_i \left(\frac{\partial u^h}{\partial x} + \frac{\partial v^h}{\partial y} \right) d\Omega = 0 \quad (2.25)$$

for equations (2.20), (2.21), and (2.22), respectively.

Employing the Galerkin formulation, the weighting functions are identified to the basis functions:

$${}^{v_x} \bar{w}_i = [N_i, 0] \quad (2.26)$$

$${}^{v_y} \bar{w}_i = [0, N_i] \quad (2.27)$$

$${}^p w_i = N_i. \quad (2.28)$$

Substituting (2.26)–(2.28), as well as the form of the approximate solution, (2.17)–(2.19), into (2.20)–(2.22) gives

$$\int_{\Omega} N_i \left[\nu \left(\frac{\partial^2}{\partial x^2} \sum_j u_j N_j \frac{\partial^2}{\partial y^2} \sum_j u_j N_j \right) - \frac{\partial}{\partial x} \sum_j p_j N_j \right] d\Omega = 0 \quad (2.29)$$

$$\int_{\Omega} N_i \left[\nu \left(\frac{\partial^2}{\partial x^2} \sum_j v_j N_j \frac{\partial^2}{\partial y^2} \sum_j v_j N_j \right) - \frac{\partial}{\partial y} \sum_j p_j N_j \right] d\Omega = 0 \quad (2.30)$$

$$\int_{\Omega} N_i \left(\frac{\partial}{\partial x} \sum_j u_j N_j + \frac{\partial}{\partial y} \sum_j v_j N_j \right) d\Omega = 0. \quad (2.31)$$

Because the basis (and weighting) functions are defined locally on each element, (2.29)–(2.31) can be re-written as the sum over the elements Ω_e of the domain Ω :

$$\sum_e \int_{\Omega_e} N_i \left[\nu \left(\frac{\partial^2}{\partial x^2} \sum_j u_j N_j \frac{\partial^2}{\partial y^2} \sum_j u_j N_j \right) - \frac{\partial}{\partial x} \sum_j p_j N_j \right] d\Omega_e = 0 \quad (2.32)$$

$$\sum_e \int_{\Omega_e} N_i \left[\nu \left(\frac{\partial^2}{\partial x^2} \sum_j v_j N_j \frac{\partial^2}{\partial y^2} \sum_j v_j N_j \right) - \frac{\partial}{\partial y} \sum_j p_j N_j \right] d\Omega_e = 0 \quad (2.33)$$

$$\sum_e \int_{\Omega_e} N_i \left(\frac{\partial}{\partial x} \sum_j u_j N_j + \frac{\partial}{\partial y} \sum_j v_j N_j \right) d\Omega_e = 0. \quad (2.34)$$

Furthermore, because the nodal values are constant, they can be taken outside of the integral. Performing this task, as well as some additional simplification and regrouping, results in the following:

$$\sum_e \sum_j \left[\left(\nu \int_{\Omega_e} N_i \left(\frac{\partial^2 N_j}{\partial x^2} + \frac{\partial^2 N_j}{\partial y^2} \right) d\Omega_e \right) u_j - \left(\int_{\Omega_e} N_i \frac{\partial N_j}{\partial x} d\Omega_e \right) p_j \right] = 0 \quad (2.35)$$

$$\sum_e \sum_j \left[\left(\nu \int_{\Omega_e} N_i \left(\frac{\partial^2 N_j}{\partial x^2} + \frac{\partial^2 N_j}{\partial y^2} \right) d\Omega_e \right) v_j - \left(\int_{\Omega_e} N_i \frac{\partial N_j}{\partial y} d\Omega_e \right) p_j \right] = 0 \quad (2.36)$$

$$\sum_e \sum_j \left[\int_{\Omega_e} \left(N_i \frac{\partial N_j}{\partial x} d\Omega_e \right) u_j + \left(\int_{\Omega_e} N_i \frac{\partial N_j}{\partial y} d\Omega_e \right) v_j \right] = 0. \quad (2.37)$$

If the integrals in (2.35)–(2.37) are evaluated, either analytically or numerically by Gauss quadrature, one obtains a linear system of equations for the unknown values u_j , v_j , and p_j as a function of their neighboring values.

Equations (2.35) and (2.36) pose a problem if linear basis functions are used. The Laplace operator results in the $\frac{\partial^2 N_j}{\partial x^2} + \frac{\partial^2 N_j}{\partial y^2}$ terms appearing in the x - and y -momentum equations. If N_j is linear, these terms are identically zero. This problem can be side-stepped by some mathematical manipulation of the weighted residual form of the equations. This will be illustrated for the x -momentum equation only—the procedure is identical for the y -momentum equation. If (2.23) is integrated by parts, one gets

$$\int_{\Omega} v_x w_i \left(\nu \nabla^2 u^h - \frac{\partial p^h}{\partial x} \right) d\Omega = \int_{\Omega} \nabla \cdot \nu v_x w_i \nabla u^h - \frac{\partial p^h}{\partial x} v_x w_i d\Omega -$$

$$\int_{\Omega} \nu \nabla^{v_x} w_i \cdot \nabla u^h - p^h \frac{\partial v_x w_i}{\partial x} d\Omega. \quad (2.38)$$

The divergence theorem states that

$$\int_{\Omega} \nabla \cdot \vec{F} d\Omega = \int_{\Gamma} \vec{F} \cdot \vec{n} d\Gamma, \quad (2.39)$$

where \vec{F} is some vector and \vec{n} is the outward unit normal on the boundary Γ of Ω . If this theorem is applied to the first integral on the right hand side of (2.38), one sees that an equivalent form of (2.23) is

$$\int_{\Gamma} \nu^{v_x} w_i \frac{\partial u^h}{\partial n} - p^h v_x w_i \frac{\partial n}{\partial x} d\Gamma - \int_{\Omega} \nu \nabla^{v_x} w_i \cdot \nabla u^h - p^h \frac{\partial v_x w_i}{\partial x} d\Omega. \quad (2.40)$$

It is seen that now all second derivatives have been removed from the equation and linear basis functions can be used.

The boundary integral term in (2.40) will be dropped for the remainder of this discussion. This can be done if certain simple physical boundary conditions are imposed.

- On Dirichlet velocity boundaries this poses no problem in that the term is not used because the velocity DOFs are prescribed on these boundaries rather than being calculated from the momentum equation. Boundaries of this type include no-slip walls and fixed-velocity inlets.
- A natural outflow boundary condition is to impose $\partial \bar{u}^h / \partial n = 0$. If we also impose that $p^h = 0$ at this boundary, the boundary integral term is identically zero anyway.

The final weighted residual FEM form of the governing Stokes equations (2.13) and (2.14) is

$$\sum_e \sum_j \left[\left(-\nu \int_{\Omega_e} \frac{\partial N_i}{\partial x} \frac{\partial N_j}{\partial x} + \frac{\partial N_i}{\partial y} \frac{\partial N_j}{\partial y} d\Omega_e \right) u_j + \left(\int_{\Omega_e} \frac{\partial N_i}{\partial x} N_j d\Omega_e \right) p_j \right] = 0 \quad (2.41)$$

$$\sum_e \sum_j \left[\left(-\nu \int_{\Omega_e} \frac{\partial N_i}{\partial x} \frac{\partial N_j}{\partial x} + \frac{\partial N_i}{\partial y} \frac{\partial N_j}{\partial y} d\Omega_e \right) v_j + \left(\int_{\Omega_e} \frac{\partial N_i}{\partial y} N_j d\Omega_e \right) p_j \right] = 0 \quad (2.42)$$

$$\sum_e \sum_j \left[\int_{\Omega_e} \left(N_i \frac{\partial N_j}{\partial x} d\Omega_e \right) u_j + \left(\int_{\Omega_e} N_i \frac{\partial N_j}{\partial y} d\Omega_e \right) v_j \right] = 0. \quad (2.43)$$

To set up a matrix equation to solve the linear system, (2.41) is used for the unknown u -DOFs, (2.42) for the v -DOFs, and (2.43) for the p -DOFs. If the typical ordering of the DOFs—first all u -DOFs, followed by all v -DOFs and then p -DOFs—is used, the resulting matrix equation has the following form:

$$\begin{bmatrix} \mathbf{K} & \mathbf{0} & {}^x\mathbf{Q} \\ \mathbf{0} & \mathbf{K} & {}^y\mathbf{Q} \\ -{}^x\mathbf{Q}^T & -{}^y\mathbf{Q}^T & \mathbf{0} \end{bmatrix} \begin{Bmatrix} \mathbf{U} \\ \mathbf{V} \\ \mathbf{P} \end{Bmatrix} = \begin{Bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \end{Bmatrix}. \quad (2.44)$$

In this equation, $\mathbf{U} = \{u_j : j = 1, 2, \dots, n\}$, $\mathbf{V} = \{v_j : j = 1, 2, \dots, n\}$, and $\mathbf{P} = \{p_j : j = 1, 2, \dots, n\}$ are the unknown DOFs, \mathbf{K} the viscous matrix, $\begin{bmatrix} {}^x\mathbf{Q} \\ {}^y\mathbf{Q} \end{bmatrix}$ the gradient matrix, and $[-{}^x\mathbf{Q}^T \ -{}^y\mathbf{Q}^T]$ the divergence matrix. Note that the system matrix is indefinite, meaning an iterative method or direct method with pivoting is required to solve the system.

In this work a node-by-node ordering of the DOFs is used, in which all velocity and pressure unknowns at a node are ordered one after the other to form a small three-component vector. This ordering results in a block-structured system matrix, a characteristic that can be exploited in the system solution algorithm. We can now speak in terms of the small characteristic 3×3 matrix equation associated with each node. The equation analogous to (2.44) is

$$\begin{bmatrix} k_{ij} & 0 & {}^xq_{ij} \\ 0 & k_{ij} & {}^yq_{ij} \\ -{}^xq_{ji} & -{}^yq_{ji} & 0 \end{bmatrix} \begin{Bmatrix} u_j \\ v_j \\ p_j \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \\ 0 \end{Bmatrix}. \quad (2.45)$$

The terms in the system matrix, which are summarized below, come from equations (2.41), (2.42), and (2.43).

$$k_{ij} = -\nu \int_{\Omega_e} \frac{\partial N_i}{\partial x} \frac{\partial N_j}{\partial x} + \frac{\partial N_i}{\partial y} \frac{\partial N_j}{\partial y} d\Omega_e \quad (2.46)$$

$${}^xq_{ij} = \int_{\Omega_e} \frac{\partial N_i}{\partial x} N_j d\Omega_e \quad (2.47)$$

$${}^yq_{ij} = \int_{\Omega_e} \frac{\partial N_i}{\partial y} N_j d\Omega_e \quad (2.48)$$

2.2.1.1 Analytical Coefficient Evaluation

Because linear triangle elements are used, each of the coefficients in (2.45) can be easily evaluated analytically from (2.46)–(2.48). These are given as

$$k_{ij} = \frac{\nu}{4S_T} ({}^x n_i {}^x n_j + {}^y n_i {}^y n_j) \quad (2.49)$$

$${}^x q_{ij} = -\frac{{}^x n_i}{6} \quad (2.50)$$

$${}^y q_{ij} = -\frac{{}^y n_i}{6} \quad (2.51)$$

In the above equations, S_T is the element area, while ${}^x n_i$ and ${}^y n_i$ are the x and y components respectively of the scaled inward element normal vector \vec{n}_i . As illustrated in Figure 2.5, the normal vector \vec{n}_i is perpendicular to, and scaled to the length of, side i (where side i is the side opposite node i).

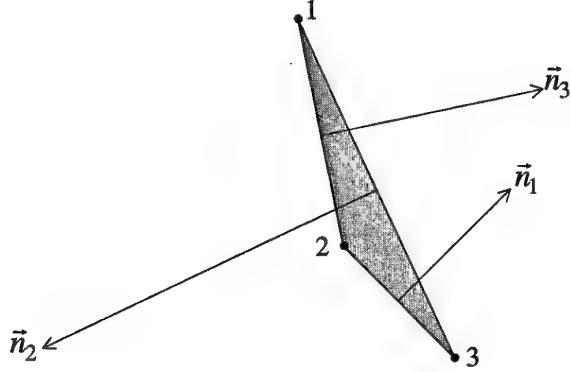


Fig. 2.5 – Definition of the scaled inward normal vectors for a linear triangle element.

2.2.1.2 Galerkin Formulation Problems

When the Galerkin FEM is applied to incompressible flow, two stability problems are encountered. First, at high Reynolds numbers (convection-dominated flow) oscillations appear in the approximate velocity solution. This phenomenon is discussed further in Sections 2.2.2.4 and 2.2.2.5. In addition, spurious pressure oscillations appear at all Reynolds numbers due to the so-called odd-even decoupling phenomenon. These instabilities are now discussed.

2.2.1.3 Spurious Pressure Oscillations

Galerkin discretizations of the Stokes Equations (and the incompressible Navier-Stokes equations, for that matter) suffer from spurious pressure oscillations in the resulting approximate solution. These oscillations occur for certain combinations of velocity and pressure basis functions. For example, Taylor and Hood (Taylor and Hood, 1973; Hood and Taylor, 1974) experienced this problem when using finite-elements with equal-order basis functions, ie. P1/P1, P2/P2, Q1/Q1, etc. No oscillations are seen for certain elements utilizing lower order basis functions for pressure than velocity, however, ie. P2/P1, P2/P0.

These above-mentioned pressure oscillations are due to the odd-even decoupling phenomenon and are seen for collocated variable discretizations in the FD and FV methods as well. This phenomenon can be easily demonstrated by considering a FD discretization on a simple 2D domain. Hypothetically, one can assume that a ‘checker-board’ pressure field has somehow developed, as shown in Figure 2.6. If a central-difference approximation is used, the pressure gradient term $\partial p / \partial x$ in the u -momentum equation is given by

$$\frac{\partial p}{\partial x} = \frac{p_{i+1,j} - p_{i-1,j}}{2\Delta x}. \quad (2.52)$$

Similarly, the pressure gradient term $\partial p / \partial y$ for the v -momentum equation is

$$\frac{\partial p}{\partial y} = \frac{p_{i,j+1} - p_{i,j-1}}{2\Delta y}. \quad (2.53)$$

The pressure at the central node (i, j) does not appear in (2.52) and (2.53). If the pressure values from the checker-board pattern in Figure 2.6 are substituted into the above equations, it is seen that all the discretized pressure gradients are zero at all the nodal points. Consequently, this pressure field would yield the same effect on the momentum equations as a uniform field, which is clearly non-physical.

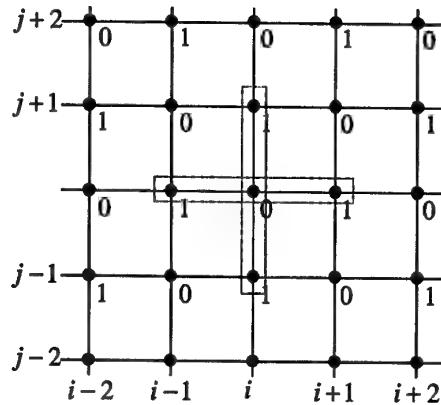


Fig. 2.6 – A hypothetical ‘checker-board’ pressure field on a uniform finite-difference mesh.

2.2.1.4 Illustration of Spurious Pressure Oscillations

The Galerkin FEM formulation of the Stokes equations using P1/P1 elements as summarized in (2.45)–(2.48) is applied to the case of a lid-driven cavity. Consider a unit square domain ($l = 1$) with no-slip walls as depicted in Figure 2.7a. The fluid is set in motion by driving the upper wall to the right at a constant velocity $U = 1$.

The mesh used in this test case is shown in Figure 2.7b. It is much finer than necessary, but will be utilized for high Reynolds number flows later. The unstructured triangle mesh contains approximately 3,000 nodes and 5,750 elements. Nodes are stretched toward the walls and clustered heavily in the corners to capture the pressure singularities that develop at the intersections of the lid and the side walls.

The resulting approximate solution is shown in Figure 2.8. The calculated velocity field, visualized by velocity vectors in Figure 2.8(a) is oscillation-free. The calculated pressure field, visualized by pressure contours in Figure 2.8(b), however, is completely polluted by spurious oscillations.

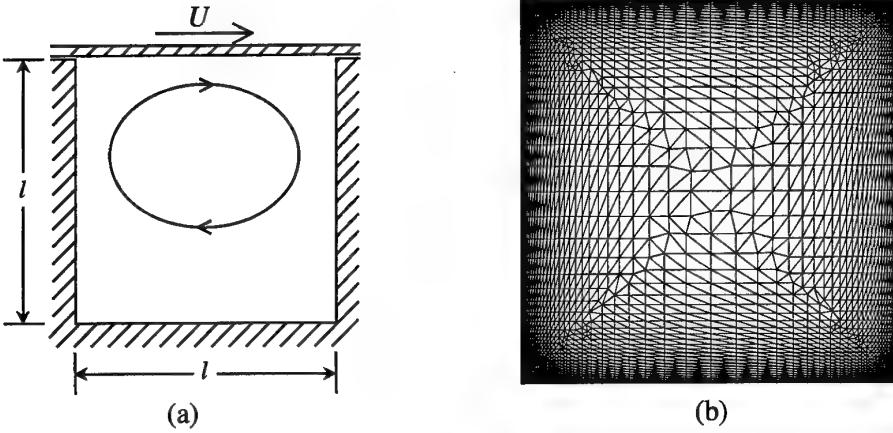


Fig. 2.7 – The lid-driven cavity test case: (a) relevant geometry and boundary conditions, and (b) unstructured triangle mesh containing approximately 3,000 nodes and 5,750 elements.

2.2.1.5 Remedyng the Spurious Pressure Oscillations

In the FE framework, the problem of spurious pressure oscillations is formulated as a violation of the so-called Babuška-Brezzi (BB) condition (Babuska, 1971, 1973; Brezzi, 1974). This mathematical compatibility condition between the discrete function spaces for the approximate pressure and velocity is rather abstract. The practical restriction placed by this condition, though, is that the pressure interpolation must be of lower order than the velocity interpolation. It should be noted that this is a necessary but not sufficient constraint to satisfy the BB condition.

The P1/P1 element used in this work, or any other equal-order element for that matter, does not satisfy the stability condition. Due to the implementation advantages of equal-order elements (mainly arising from a much simpler data structure), approaches to circumvent the BB condition on this type of element have been developed in recent years. The approach used in this work is the Pressure Stabilized/Petrov-Galerkin (PSPG) method originally proposed by Hughes et al. in 1986 (Hughes et al., 1986) and refined by researchers such as Tezduyar (Tezduyar et al., 1991, 1992) and Franca (Franca and Frey, 1992).

2.2.2 PSPG Formulation

The PSPG formulation is a full Petrov-Galerkin discretization of the momentum/continuity system. If we write the Galerkin vector weight functions as

$$\vec{W}^h = ({}^v\vec{w}^h, {}^p w^h), \quad (2.54)$$

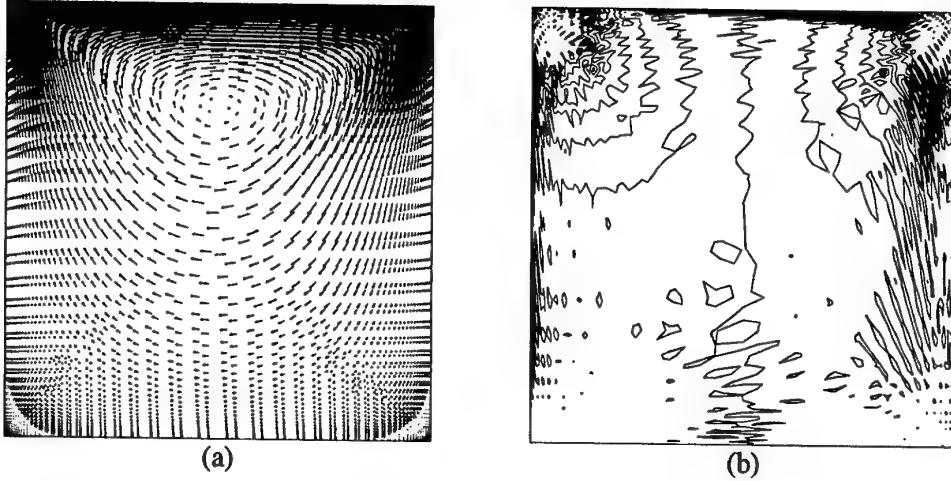


Fig. 2.8 – Stokes flow lid-driven cavity Galerkin results: (a) calculated velocity vectors, and (b) calculated pressure field corrupted by spurious oscillations.

the PSPG perturbed weight function applied to the system is

$$\vec{W}_{PSPG}^h = ({}^v\vec{w}^h, {}^p w^h) + \tau_e (\nabla {}^p w^h, 0). \quad (2.55)$$

The PSPG method can also be interpreted as a stabilized FE formulation in which a stabilization term is added to the Galerkin discretization of the continuity equation to eliminate the spurious pressure modes resulting from the violation of the BB condition. The form of the stabilization term is

$$S_{PSPG} = \sum_e \int_{\Omega_e} (\tau_e \nabla {}^p w^h) \cdot \vec{R}^h d\Omega, \quad (2.56)$$

where \vec{R}^h is the residual of the discrete momentum equation, or

$$\vec{R}^h = \nabla p^h - \nu \nabla^2 \vec{u}^h \quad (2.57)$$

for Stokes flow with no body forces. This term enhances stability without upsetting accuracy or consistency because as the solution converges $\vec{R}^h \rightarrow 0$ and consequently $S \rightarrow 0$.

The stabilization term S also contains an intrinsic time scale τ_e . This time scale is usually defined on an elemental level and in an ad-hoc way. Convergence analysis shows that τ_e should be $\mathcal{O}(h_e^2/\nu)$, where h_e is the hydraulic diameter of the element,

$$h_e = \sqrt{\frac{4S_T}{\pi}}. \quad (2.58)$$

Hughes, et al. (Hughes et al., 1986) found empirically that if τ_e is defined as

$$\tau_e = \alpha \frac{h_e^2}{2\nu}, \quad (2.59)$$

the spurious pressure modes are suppressed if $\alpha \geq 0.1$ in the case of Q1/Q1 elements. Tezduyar (Tezduyar et al., 1991) recommends τ_e be a function of an element Reynolds number Re^U ,

$$\tau_e = \frac{h_e}{2U} \zeta(Re^U). \quad (2.60)$$

The characteristic element Reynolds number Re^U is based on a velocity U and characteristic element length h_e :

$$Re^U = \frac{h_e U}{2\nu}. \quad (2.61)$$

Tezduyar uses a global flow velocity for U rather than the local velocity at each element. This is necessary because PSPG stabilization is required at all points in the flow, even where the local Reynolds number is zero. There are numerous formulations of the ζ function proposed in the literature (see DeMulder (DeMulder et al., 1994)). All of these ζ functions have the property of being $\mathcal{O}(0)$ for $Re \rightarrow 0$ and $\mathcal{O}(1)$ for $Re \rightarrow \infty$. In SFE2D, the form of τ_e proposed by Tezduyar (2.60) is employed.

In terms of the nodal matrix equation, the PSPG formulation contains additional terms in the continuity equation, including a term on the diagonal. Specifically, (2.45) becomes

$$\begin{bmatrix} k_{ij} & 0 & {}^x q_{ij} \\ 0 & k_{ij} & {}^y q_{ij} \\ -{}^x q_{ji} + {}^{kx} S_{Pij} & -{}^y q_{ji} + {}^{ky} S_{Pij} & {}^q S_{Pij} \end{bmatrix} \begin{Bmatrix} u_j \\ v_j \\ p_j \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \\ 0 \end{Bmatrix}. \quad (2.62)$$

The new (stabilization) terms appearing in (2.62) that are not in (2.45) are evaluated from (2.56) as

$${}^{kx} S_{Pij} = \int_{\Omega_e} \tau_e \frac{\partial N_i}{\partial x} \nabla^2 N_j d\Omega_e \quad (2.63)$$

$${}^{ky} S_{Pij} = \int_{\Omega_e} \tau_e \frac{\partial N_i}{\partial y} \nabla^2 N_j d\Omega_e \quad (2.64)$$

$${}^q S_{Pij} = \int_{\Omega_e} \tau_e \frac{\partial N_i}{\partial x} \frac{\partial N_j}{\partial x} d\Omega_e + \int_{\Omega_e} \tau_e \frac{\partial N_i}{\partial y} \frac{\partial N_j}{\partial y} d\Omega_e \quad (2.65)$$

2.2.2.1 Analytical Coefficient Evaluation

Because linear triangle elements are used, each of the stabilization coefficients in (2.62) can be evaluated analytically from (2.63)–(2.65), and are given as

$${}^{kx} S_{Pij} = 0 \quad (2.66)$$

$$k_y S_{Pij} = 0 \quad (2.67)$$

$$q S_{Pij} = \frac{\tau_e}{S_T} ({}^x n_i {}^x n_j + {}^y n_i {}^y n_j). \quad (2.68)$$

The stabilization terms $k_x S_{Pij}$ and $k_y S_{Pij}$ are zero due to the inability of linear elements to possess a non-zero second derivative, i.e.

$$\frac{\partial^2 N_i}{\partial x^2} = \frac{\partial^2 N_i}{\partial y^2} = 0. \quad (2.69)$$

2.2.2.2 Illustration of PSPG Stokes Solution

The PSPG FE formulation of the Stokes equations, as summarized in (2.62), is applied to the lid-driven cavity test case described in Section 2.2.1.4 to show the effects of the pressure stabilization. The resulting approximate solution is shown in Figure 2.9. The calculated velocity field, visualized by velocity vectors in Figure 2.9(a), is virtually identical to that obtained using the Galerkin approach, shown in Figure 2.8(a). However, the pressure field, visualized by pressure contours in Figure 2.9(b), is now seen to be smooth and oscillation free.

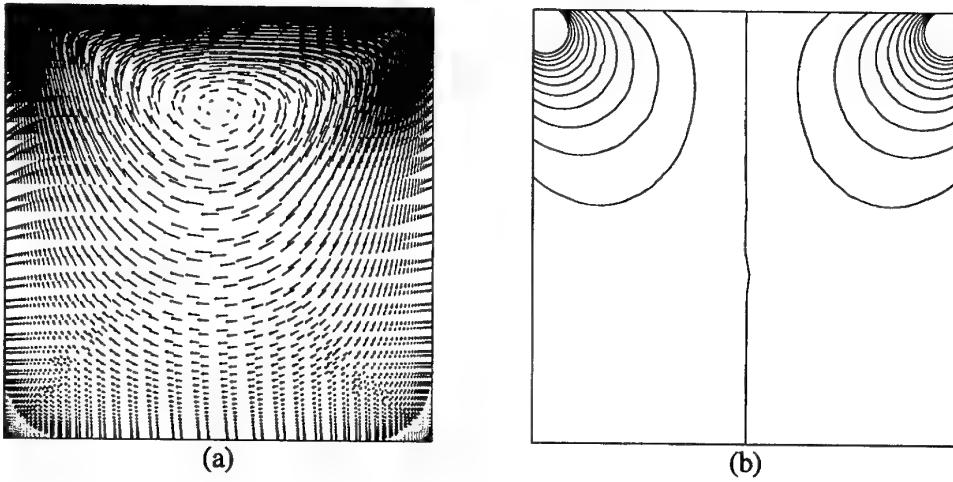


Fig. 2.9 – Stokes flow lid-driven cavity PSPG results: (a) calculated velocity vectors, and (b) calculated pressure field (25 contours over the range $-20 \leq p \leq 20$).

2.2.2.3 PSPG Formulation Problems

PSPG FE solutions can still suffer from convection-induced oscillations in the velocity field. As mentioned previously, the Galerkin (and PSPG) FEM gives rise to central-difference type approximations of the differential operators. As expected, these convection-induced oscillations also plague central-difference FDMs and FVMs.

2.2.2.4 Illustration of Convection-Induced Oscillations

To illustrate the convection-induced oscillations in the PSPG solution, the method previously described is applied to the case of flow past a square block in a channel. The Reynolds number, based on block height and average inlet velocity, is 200. Figure 2.10 shows the resulting velocity field near the block computed on a coarse mesh with a constant node spacing of 0.5 block heights. It is seen that the velocity solution contains spurious oscillations, rendering the approximation useless.

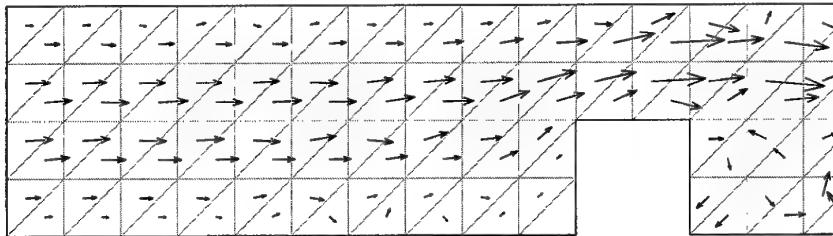


Fig. 2.10 – Velocity vectors showing convection-induced oscillations in the PSPG solution of flow past a square block at $Re = 200$.

2.2.2.5 Remedyng the Convection-Induced Oscillations

The convection-induced oscillations can be handled in two ways. The first is to keep the element Peclet number,

$$P_e = \frac{\rho u h_e}{\nu}, \quad (2.70)$$

small ($P_e < 2$ guarantees stability).

The other option is to introduce certain forms of upwinding (Gresho and Lee, 1979; Versteeg and Malalasekera, 1995). A number of upwinding schemes have been developed in the FE framework (Christie et al., 1976; Heinrich et al., 1977; Hughes, 1978; Hughes and Atkinson, 1978), the most common being the Streamline-Upwind Petrov-Galerkin (SUPG) formulation (Brooks and Hughes, 1982; Tezduyar and Hughes, 1983; Mizukami, 1985). This formulation is advantageous because it is consistent and not subject to the artificial cross-flow diffusion criticisms associated with many classical upwind methods.

The SUPG method increases control over the convective terms by adding artificial diffusion acting only in the streamline direction. If we write the Galerkin vector weight function as

$$\vec{W}^h = ({}^v \bar{w}^h, {}^p w^h), \quad (2.71)$$

the SUPG perturbed weight function is

$$\vec{W}_{SUPG}^h = (\vec{w}^h, p_w^h) + \tau_{SUPG} (\vec{u}^h \cdot \nabla \vec{w}^h, 0). \quad (2.72)$$

The SUPG method can also be interpreted as a stabilized FE formulation in which a stabilization term is added to the Galerkin discretization of the momentum equation. The form of the stabilization term is

$$S_{SUPG} = \sum_e \int_{\Omega_e} (\tau_{SUPG} \vec{u}^h \cdot \nabla \vec{w}^h) \cdot \vec{R}_{NS}^h d\Omega, \quad (2.73)$$

where \vec{R}_{NS}^h is the residual of the discrete NS momentum equation.

The ‘intrinsic time scale’ τ_{SUPG} is defined as

$$\tau_{SUPG} = \frac{h}{2\|\vec{u}^h\|} \zeta(Re_u), \quad (2.74)$$

where the local element Reynolds number, Re_u is based on the local velocity \vec{u}^h as

$$Re_u = \frac{h\|\vec{u}^h\|}{2\nu} \quad (2.75)$$

and

$$\|\vec{u}^h\| = \sqrt{u^{h2} + v^{h2}}. \quad (2.76)$$

Instead of the hydraulic diameter, the characteristic length used for τ_{SUPG} is the element length in the direction of the local flow, or

$$h = 2 \left(\sum_{i=1}^{NPE} |\vec{s} \cdot \nabla N_i| \right), \quad (2.77)$$

where NPE is the number of nodes per element, \vec{s} is the unit vector in the direction of the local velocity, and N_i is the basis function associated with node i .

The ζ function possibilities are the same as those for the PSPG method and, again, are well presented by DeMulder et al. (DeMulder et al., 1994). For this solver ζ is defined as

$$\zeta(Re) = \begin{cases} Re/3, & \text{for } 0 \leq Re \leq 3, \\ 1, & \text{for } 3 \leq Re. \end{cases} \quad (2.78)$$

The addition of this SUPG stabilization term has the same effect as adding upwinding in the FDM and FVM. That is, the values at upstream nodes are weighted more heavily than downstream nodes. This is illustrated in Figure 2.11, which shows typical Galerkin and SUPG weighting functions in one dimension.

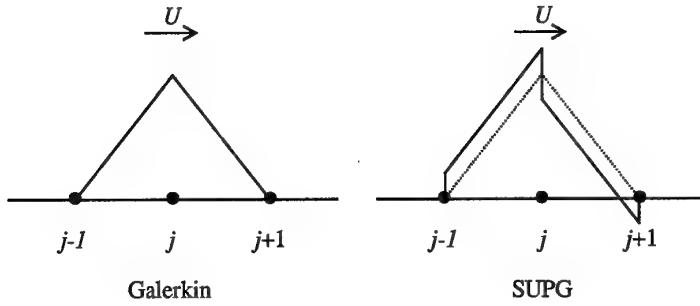


Fig. 2.11 – Comparison of the weighting functions for the Galerkin and SUPG methods. The Galerkin method weights upstream and downstream nodes equally, whereas the SUPG method weights the upstream nodes more heavily. The dashed line in the SUPG method shows the basis functions are unchanged from the Galerkin method.

In Figures 2.12 and 2.13, numerical results are again shown for the test case of flow over a square bump. The results in Figure 2.12 are obtained by employing a SUPG/PSPG formulation utilizing the same mesh mentioned previously (with constant node spacing of 0.5 block heights). It is seen that the addition of the upwinding terms has removed the convective instabilities. In Figure 2.13, on the other hand, no upwinding is used. Instead, the mesh has been refined sufficiently to remove the convection-induced oscillations. The mesh used to obtain these oscillation-free results has a node spacing of 0.1 block heights.

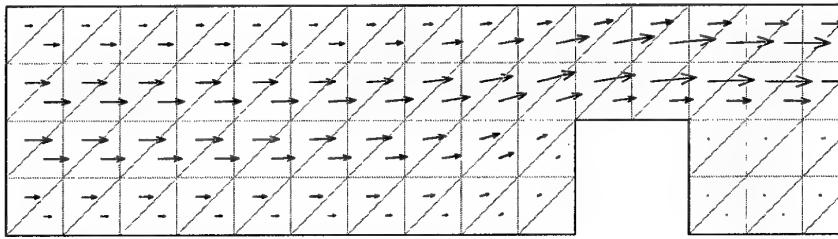


Fig. 2.12 – Velocity vectors showing the oscillation-free SUPG/PSPG coarse-mesh velocity solution of flow past a square block at $Re = 200$.

2.3 Temporal Discretization

In this section the temporal discretization is described. Because the convective terms are treated explicitly in time, this is the first point at which the full incompressible NS equations can be considered.

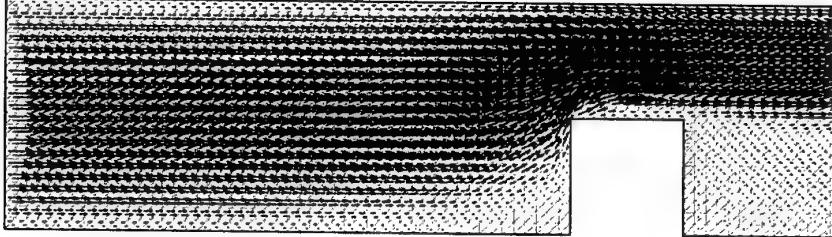


Fig. 2.13 – Velocity vectors showing the oscillation-free PSPG fine-mesh velocity solution of flow past a square block at $Re = 200$.

Consider, then, incompressible laminar flow of a Newtonian fluid in the computational domain Ω with boundary $\Gamma = \Gamma_d \cup \Gamma_n$, where Γ_d is the portion of the boundary subject to Dirichlet boundary conditions and Γ_n the portion subject to Neumann conditions. The flow is governed by the momentum equation

$$\frac{\partial \vec{u}}{\partial t} + (\vec{u} \cdot \nabla) \vec{u} + \nabla p - \nu \nabla^2 \vec{u} = 0 \quad (2.79)$$

and the continuity equation

$$\nabla \cdot \vec{u} = 0. \quad (2.80)$$

In the above equations, ν is the kinematic viscosity of the fluid, \vec{u} is the 2D velocity vector (u, v) , and p is the kinematic pressure (pressure divided by density).

Equations (2.79) and (2.80) are subject to Dirichlet and Neumann boundary conditions:

$$\vec{u} = \bar{\vec{u}} \quad \text{on } \Gamma_d, \quad (2.81)$$

$$(-p[I] + \nu \nabla \vec{u}) \cdot \vec{n} = \bar{\theta} \quad \text{on } \Gamma_n, \quad (2.82)$$

where $\bar{\vec{u}}$ is the imposed Dirichlet velocity boundary condition, $[I]$ is the identity matrix, \vec{n} is the outward unit normal on Γ , and $\bar{\theta}$ is the imposed surface traction boundary condition.

2.3.1 Pressure and Diffusion Terms

It is desired to have second-order accuracy in the time discretization. For the pressure and diffusion terms, this is accomplished by applying a Crank-Nicolson method. In terms of a simple example first-order ODE

$$\frac{df}{dt} = \lambda f, \quad (2.83)$$

this method results in the discrete expression

$$\frac{f^{n+1} - f^n}{\Delta t} = \lambda \frac{f^{n+1} + f^n}{2}, \quad (2.84)$$

where $[]^{n+1}$ denotes the new time step and $[]^n$ the current time step.

It is apparent from (2.84) that the Crank-Nicolson method is based on central differencing (centered about time $n + \frac{1}{2}$) and hence is second-order accurate. Due to the implicit nature of the scheme, it is mathematically deemed unconditionally stable for all time step values, meaning that perturbations are not amplified with time. In practice, though, it is found that oscillations can occur for sufficiently large time step values.

2.3.2 Convective Terms

The convective terms are treated using a second-order explicit Adams-Bashforth method. Again, in terms of the example first-order ODE (2.83), the method results in the discrete expression

$$\frac{f^{n+1} - f^n}{\Delta t} = \lambda \left(\frac{3}{2} f^n - \frac{1}{2} f^{n-1} \right). \quad (2.85)$$

As with all explicit schemes, the Adams-Bashforth method imposes a stability limit on the time step size. Though this can be a serious limitation in some cases, an explicit treatment of the convection terms is desirable in this work for two main reasons.

- The convective terms are non-linear. Therefore, if treated explicitly, the system is linear within each time step and there is no need for multiple iterations within the time step or the necessity to calculate expensive Jacobian matrices.
- The ultimate intent of this solver is to perform LES. Since LES requires very small time steps anyway, the time step limitation due to the explicit convection treatment is less important.

2.3.3 SUPG/PSPG Formulation of the NS Equations

When the previously mentioned time discretizations are applied to the 2D incompressible NS equations, the following expression is obtained:

$$\begin{aligned} \frac{\vec{u}^{n+1} - \vec{u}^n}{\Delta t} + \frac{3}{2} (\vec{u}^n \cdot \nabla) \vec{u}^n - \frac{1}{2} (\vec{u}^{n-1} \cdot \nabla) \vec{u}^{n-1} = \\ - \frac{1}{2} (\nabla p^{n+1} + \nabla p^n) + \frac{1}{2} \nu (\nabla^2 \vec{u}^{n+1} + \nabla^2 \vec{u}^n) \end{aligned} \quad (2.86)$$

Applying the SUPG/PSPG FEM to the above equation results in a matrix equation to be solved at each time step to obtain the approximate solution. To assist in implementation, the matrix equation is written in terms of the change in flow parameters during each time step,

$$\delta \vec{u} = \vec{u}^{n+1} - \vec{u}^n, \quad (2.87)$$

$$\delta p = p^{n+1} - p^n, \quad (2.88)$$

rather than the flow parameters themselves, \bar{u}^{n+1} and p^{n+1} . The resulting nodal matrix equation is

$$\begin{aligned} & \left[\begin{array}{ccc} \frac{1}{\Delta t} m_{ij} + \frac{1}{2} k_{ij} & 0 & \frac{1}{2} {}^x q_{ij} \\ 0 & \frac{1}{\Delta t} m_{ij} + \frac{1}{2} k_{ij} & \frac{1}{2} {}^y q_{ij} \\ - {}^x q_{ji} + \frac{1}{\Delta t} {}^{m_x} S_{Pij} + \frac{1}{2} k_x {}^x S_{Pij} & - {}^y q_{ji} + \frac{1}{\Delta t} {}^{m_y} S_{Pij} + \frac{1}{2} k_y {}^y S_{Pij} & \frac{1}{2} {}^q S_{Pij} \end{array} \right] \begin{Bmatrix} \delta u_j \\ \delta v_j \\ \delta p_j \end{Bmatrix} = \\ & - \left[\begin{array}{ccc} k_{ij} & 0 & {}^x q_{ij} \\ 0 & k_{ij} & {}^y q_{ij} \\ - {}^x q_{ji} + {}^{k_x} S_{Pij} & - {}^y q_{ji} + {}^{k_y} S_{Pij} & {}^q S_{Pij} \end{array} \right] \begin{Bmatrix} u_j \\ v_j \\ p_j \end{Bmatrix}^n \\ & - \frac{3}{2} \begin{Bmatrix} {}^x c_j + {}^x S_{Uj} \\ {}^y c_j + {}^y S_{Uj} \\ {}^q S_{Pj} \end{Bmatrix}^n + \frac{1}{2} \begin{Bmatrix} {}^x c_j + {}^x S_{Uj} \\ {}^y c_j + {}^y S_{Uj} \\ {}^q S_{Pj} \end{Bmatrix}^{n-1}. \quad (2.89) \end{aligned}$$

All SUPG terms are treated in an explicit manner for the same reason as the convective terms. That is, because they contain a product of unknown velocities, the terms are nonlinear. Because all nonlinear terms are treated explicitly, the terms on the right-hand-side are all geometry-based, and are constant for a given mesh (assuming the boundary conditions are also constant).

Without explicitly writing the complete derivation, the new terms in the matrix equation are summarized below.

$$m_{ij} = \int_{\Omega_e} N_i N_j d\Omega_e \quad (2.90)$$

$${}^{m_x} S_{Pij} = \int_{\Omega_e} \tau_e \frac{\partial N_i}{\partial x} N_j d\Omega_e \quad (2.91)$$

$${}^{m_y} S_{Pij} = \int_{\Omega_e} \tau_e \frac{\partial N_i}{\partial y} N_j d\Omega_e \quad (2.92)$$

$${}^x c_i = \int_{\Omega_e} N_i \left(u_k N_k \frac{\partial u_j N_j}{\partial x} + v_k N_k \frac{\partial v_j N_j}{\partial y} \right) d\Omega_e \quad (2.93)$$

$${}^y c_i = \int_{\Omega_e} N_i \left(u_k N_k \frac{\partial v_j N_j}{\partial x} + v_k N_k \frac{\partial u_j N_j}{\partial y} \right) d\Omega_e \quad (2.94)$$

$$\begin{aligned} {}^q S_{Pj} = & \int_{\Omega_e} \tau_e \frac{\partial N_i}{\partial x} \left(u_k N_k \frac{\partial u_j N_j}{\partial x} + v_k N_k \frac{\partial v_j N_j}{\partial y} \right) + \\ & \frac{\partial N_i}{\partial y} \left(u_k N_k \frac{\partial v_j N_j}{\partial x} + v_k N_k \frac{\partial u_j N_j}{\partial y} \right) d\Omega_e \end{aligned} \quad (2.95)$$

$${}^x S_{U,i} = \int_{\Omega_e} \tau_{SUPG} \left(\hat{u} \frac{\partial N_i}{\partial x} + \hat{v} \frac{\partial N_i}{\partial y} \right) \left[\frac{\Delta u_j}{\Delta t} N_j + u_j \left(\hat{u} \frac{\partial N_j}{\partial x} + \hat{v} \frac{\partial N_j}{\partial y} \right) + p_j \frac{\partial N_j}{\partial x} \right] d\Omega_e \quad (2.96)$$

$${}^y S_{U,i} = \int_{\Omega_e} \tau_{SUPG} \left(\hat{u} \frac{\partial N_i}{\partial x} + \hat{v} \frac{\partial N_i}{\partial y} \right) \left[\frac{\Delta v_j}{\Delta t} N_j + v_j \left(\hat{u} \frac{\partial N_j}{\partial x} + \hat{v} \frac{\partial N_j}{\partial y} \right) + p_j \frac{\partial N_j}{\partial y} \right] d\Omega_e \quad (2.97)$$

2.3.3.1 Analytical Coefficient Evaluation

Because linear triangle elements are used, each of the coefficients in (2.89) can be evaluated analytically. The new terms introduced in this section, (2.90)–(2.95), become

$$m_{ij} = \frac{1 + \delta_{ij}}{12} S_T \quad (2.98)$$

$${}^{mx} S_{P,ij} = \tau_e \frac{{}^x n_i}{6} \quad (2.99)$$

$${}^{my} S_{P,ij} = \tau_e \frac{{}^y n_i}{6} \quad (2.100)$$

$${}^x C_i = \frac{1 + \delta_{ij}}{24} u_j (u_k {}^x n_j + v_k {}^y n_j) \quad (2.101)$$

$${}^y C_i = \frac{1 + \delta_{ij}}{24} v_j (u_k {}^x n_j + v_k {}^y n_j) \quad (2.102)$$

$${}^c S_{P,i} = \frac{\tau_e}{12 S_T} (u_j {}^x n_i + v_j {}^y n_i) (u_k {}^x n_j + v_k {}^y n_j) \quad (2.103)$$

$${}^x S_{U,j} = \tau_{SUPG} \left(\frac{1 + \delta_{jk}}{24} \frac{\Delta u_j}{\Delta t} + \frac{1}{12 S_T} p_j {}^x n_j + \frac{1}{12 S_T} u_j (u_k {}^x n_j + v_k {}^y n_j) \right) (u_k {}^x n_i + v_k {}^y n_i) \quad (2.104)$$

$${}^y S_{U,j} = \tau_{SUPG} \left(\frac{1 + \delta_{jk}}{24} \frac{\Delta v_j}{\Delta t} + \frac{1}{12 S_T} p_j {}^y n_j + \frac{1}{12 S_T} v_j (u_k {}^x n_j + v_k {}^y n_j) \right) (u_k {}^x n_i + v_k {}^y n_i), \quad (2.105)$$

where δ is the Kronecker delta operator.

2.4 Linear System Solution

The SPARSKIT tool kit, developed by Youcef Saad, University of Minnesota (Saad, 1994b), was used to solve the linear system of algebraic equations generated by the SUPG/PSPG FEM. The kit includes a number of routines written in Fortran77 for manipulating and work-

ing with sparse matrices, including iterative algorithms for solving the matrix equation

$$[A]\{x\} = \{b\}, \quad (2.106)$$

where $[A]$ is a user-supplied $n \times n$ sparse matrix, $\{b\}$ is a user-supplied vector of length n , and $\{x\}$ is a vector of unknowns of length n to be computed.

SPARSKIT includes a number of Krylov iterative methods (Saad, 1981) used in conjunction with various preconditioners. In particular, satisfactory results in terms of robustness and speed have been obtained using a restarted GMRES algorithm (Saad and Schultz, 1986) with an ILUT preconditioner (Saad, 1994a).

A selection of sparse matrix storage formats are available to describe the system matrix. Though it does not exploit the block structure of the matrix, the compressed sparse row format (Appendix A) is used inside SFE2D because it is the format SPARSKIT uses for its internal computations.

2.5 SFE2D Summary

This section provides a detailed summary of the implementation of the SUPG/PSPG FEM in the SFE2D NS solver.

The solver finds an approximate solution to the 2D unsteady NS equations in terms of the primitive variables \vec{u} , P . When the governing equations (2.79)–(2.80) are discretized, a system of linear equations is formed that must be solved to obtain the approximate solution. Because the SFE2D solver uses a nodal ordering of the unknowns, when this linear system is written in matrix form, the matrix equation to be solved (2.89) can be characterized by a 3×3 system associated with each node, (2.89).

All the unknowns appearing in (2.89) are obtained by looping through the nodes of each element and summing the contributions. Each of the coefficient contributions are derived from analytical expressions derived from the PSPG FEM. These expressions are given in (2.49)–(2.51), (2.66)–(2.68), and (2.98)–(2.103).

Figure 2.14 presents a flow chart giving the basic structure of the SFE2D solver. Each of the blocks in this chart will now be discussed in more detail.

2.5.0.1.1 Initialize.

The user-defined parameters for SFE2D are given via a Fortran NAMELIST file. These parameters contain such information as time step size, number of time steps, convergence criteria, sparse matrix solver parameters, and Dirichlet boundary condition values. Additionally, input and output filenames are obtained and the corresponding files are opened.

2.5.0.1.2 Read grid and flow variables.

The input grid file is read. The format of this file is the DPlot format developed at the von Karman Institute and is described in Appendix B. This file contains the 2D nodal coordinates and connectivity as well as flow values at each node.

2.5.0.1.3 Create CSR matrix structure from the connectivity information.

The structure of the sparse system matrix (the location of the non-zero entries) can be determined from the mesh connectivity information read from the DPlot input file. This structure is stored in the ia and ja arrays as described in Appendix A.

2.5.0.1.4 Set initial conditions.

The initial conditions—the values of \bar{u} and p for levels n and $n-1$ at the first time step—are set from the values stored in the input DPlot file if a restart is requested. Otherwise, the initial conditions are set to zero.

2.5.0.1.5 Build $[A]$ matrix from SUPG/PSPG FEM and boundary conditions...store.

The non-zero values in the system matrix are calculated from (2.89) and the boundary conditions. All values in this matrix are geometry-based and do not change from time step to time step (assuming boundary condition types do not change). Consequently, this matrix is calculated only once and stored for use in all time steps.

2.5.0.1.6 Create $\{b\}$ vector from SUPG/PSPG FEM and boundary conditions.

The right-hand-side vector is calculated from (2.89) and the boundary conditions. This vector is calculated at each time step.

2.5.0.1.7 Solve $[A]\{x\} = \{b\}$ using SPARSKIT.

The matrix equation is passed to the SPARSKIT routines to be solved. Because an iterative method is used, an initial guess for the solution is required. SFE2D uses the zero vector as the initial guess because $\{x\}$ contains the change in flow variables between time steps and the time steps are assumed small in order to satisfy the stability criteria of the explicit convection treatment. Because the system matrix is constant, the preconditioning matrix is calculated only at the first time step and stored for use on all other time steps.

2.5.0.1.8 Update variables for next time step.

Once the matrix equation has been approximately solved, the flow variables are updated. That is,

$$\begin{aligned}\vec{u}_i^{n-1} &= \vec{u}_i^n, & p_i^{n-1} &= p_i^n \\ \vec{u}_i^n &= \vec{u}_i^{n+1}, & p_i^n &= p_i^{n+1} \\ \vec{u}_i^{n+1} &= \vec{u}_i^{n+1} + \delta\vec{u}_i, & p_i^{n+1} &= p_i^{n+1} + \delta p_i\end{aligned}$$

2.5.0.1.9 Output to file if requested.

The Fortran NAMELIST file contains an output frequency parameter stating the number of time steps between writing of the output files. If the current time step requests output files to be written, a ‘restart’ file is written in DPlot format as well as a post-processing file in TecPlot and/or FieldView unstructured grid format.

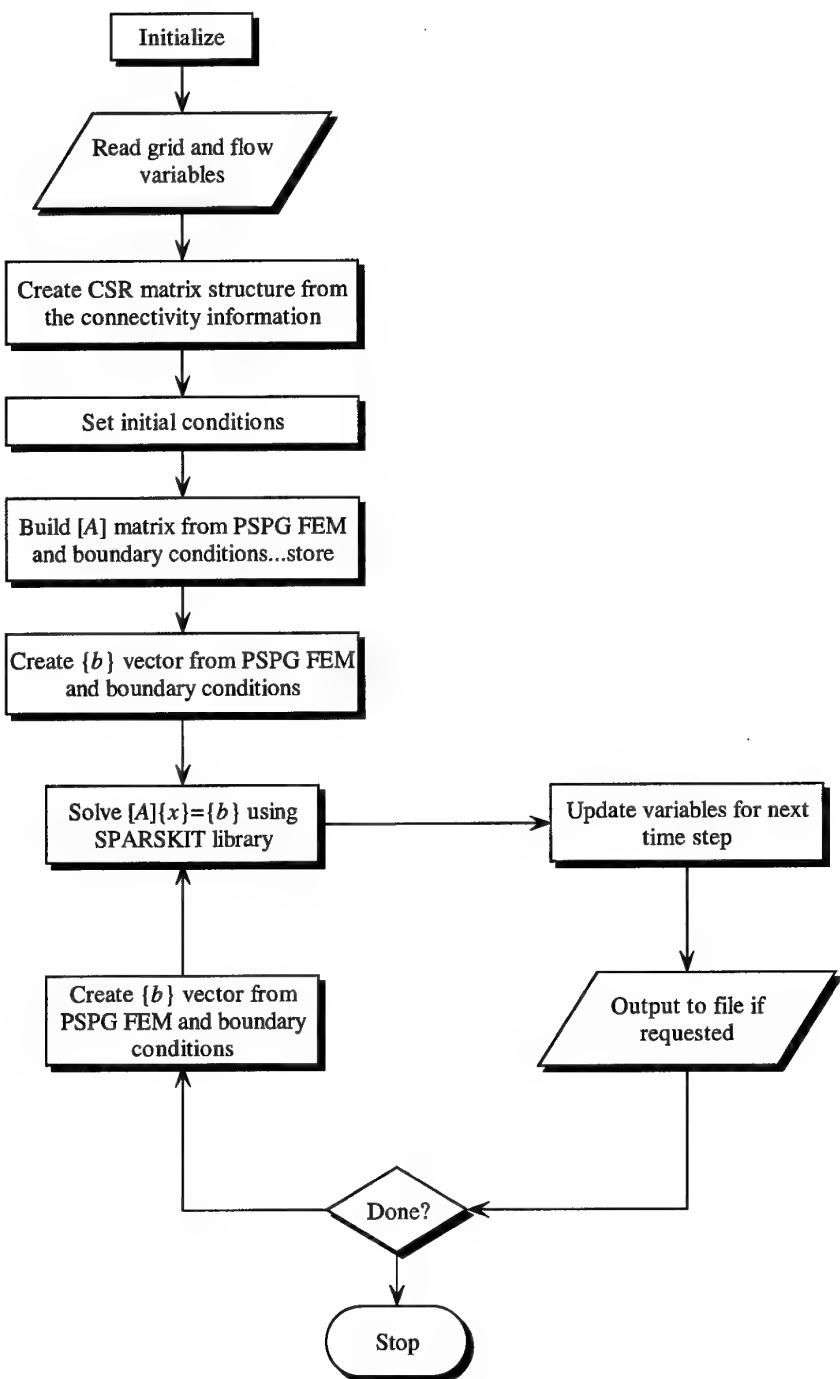


Fig. 2.14 – Flow chart showing the basic structure of the SFE2D Navier-Stokes solver.

3. 2D LAMINAR FLOW TEST CASES

This chapter presents a number of test cases showing the validity of the SFE2D solver. The first case presented is that of steady lid-driven cavity flow. Next, steady flow past a backward-facing step is calculated. Finally, time-accurate results for flow past a circular cylinder in a channel are presented.

3.1 Lid-Driven Flow in a Square Cavity

3.1.1 Problem Description

In this first test case, flow inside a square cavity as depicted in Figure 3.1 is considered. The cavity has unit dimensions ($l = 1$) and the flow is set in motion by driving the top wall to the right at a constant velocity ($U = 1$). No-slip velocity conditions are applied on all boundaries, resulting in a purely Dirichlet velocity problem. Because the pressure field need only be defined up to an arbitrary constant, a pressure datum $p = 0$ is set at the center of the top wall.

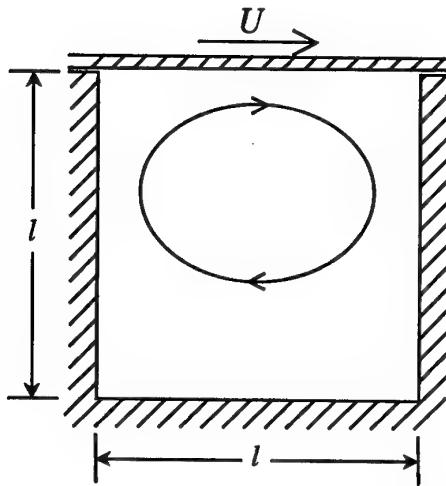


Fig. 3.1 – Relevant geometry and boundary conditions for the lid-driven cavity test case.

Although this case is purely academic in nature, it is useful in testing numerical algorithms for solving the NS equations for a number of reasons. First, the boundary conditions are well defined and easy to implement. There are no concerns about whether inflow/outflow conditions are appropriately set, and the extents of the computational domain are clearly defined. In addition, a number of different flow regimes are contained in the flow: a wall

boundary layer, an internal shear layer, and recirculating flow. Finally, the properties of the flowfield vary distinctly with Reynolds number. The main characteristics of the flowfield are a primary recirculation cell near the center of the cavity and pressure singularities at the top corners where there is a discontinuity in the velocity boundary conditions. At high Reynolds numbers, secondary and tertiary vortices develop in the corners.

The results of SFE2D are compared to the benchmark solutions of Ghia et al. (Ghia et al., 1988). Ghia's results were obtained in 1982 using a second-order finite-difference approach on very fine, uniform Cartesian meshes and have been accepted as benchmark solutions for many years. Typical mesh sizes used by Ghia are $129 \times 129 = 16641$ nodes and $257 \times 257 = 66049$ nodes. By proper mesh design, SFE2D produces results similar in accuracy using far fewer nodes.

As explained in Chapter 2, SFE2D is an unsteady solver. Consequently, the steady-state solutions presented here are obtained by starting with a stagnant flowfield and marching in time until the flow no longer changes. Surely, SFE2D is not efficient at solving these steady-state problems because it was designed specifically for time-accurate simulations. However, the lid-driven cavity test case allows the spatial accuracy of the algorithm to be examined.

3.1.2 Computational Mesh

The computational mesh is shown in Figure 3.2. It was constructed using the Delaundo grid generator developed by J.D. Müller at the von Karman Institute. This grid generator is based on ideas from both the Delaunay triangulation approach and the advancing front method (Müller et al., 1993), resulting in good connectivity, smooth point distributions, and efficient implementation. The user supplies the boundary geometry and point distribution, and Delaundo creates an unstructured mesh of approximately isotropic triangles.

Near solid boundaries it is often necessary to stretch the nodes close to the wall in order to resolve the boundary layer. If this is done with isotropic triangles, Figure 3.3(a), the node spacing tangent to the wall is approximately the same as in the normal direction. This results in a terrible waste of nodes because the gradients tangent to the wall are much smaller than the normal gradients. What is preferred is to use high aspect-ratio elements in the boundary layer. Using standard unstructured triangle algorithms in these boundary layer regions result in awkward triangles similar to that shown in Figure 3.3(b), which are poorly-suited for numerical computations. Babuška and Aziz (Babuska and Aziz, 1976) have proved that the quality of a piecewise-linear FE approximation degrades for high-aspect ratio cell only if the maximum angle is too near 180 degrees. Consequently, structured 'wedges,' as shown in Figure 3.3(c) are used in the boundary layer regions.

The mesh shown in Figure 3.2 consists of 3013 nodes and 5752 linear triangle elements. Boundary-layer regions of structured wedges exist along each of the four walls. Very near the corners (within 0.025 units), the wedge layers give way to very small isotropic triangles

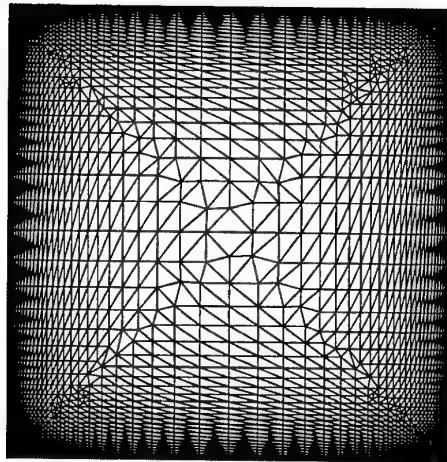


Fig. 3.2 – Unstructured triangle mesh utilized for the lid-driven cavity test case, containing 3013 nodes and 5752 elements.

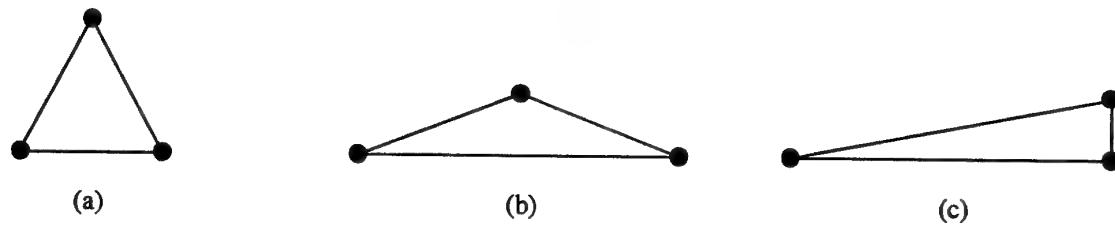


Fig. 3.3 – Triangle elements: (a) isotropic triangle, (b) high aspect ratio triangle poorly suited for numerical computations, (c) high aspect ratio ‘wedge’ well suited for numerical computations.

(Figure 3.4). This allows for very tight clustering near the corners in order to better capture the pressure singularities that may exist.

3.1.3 $Re = 100$

A particle tracer visualization of the velocity field for $Re = 100$ is shown in Figure 3.5. This figure, as well as the velocity vectors in Figure 3.6, show the primary vortex core is located toward the upper-right corner of the cavity ($x, y) = (0.623, 0.741)$) at this Reynolds number. Also, very weak vortices appear in the bottom corners. The associated pressure field is displayed via contour lines in Figure 3.7, where singularities can be seen in the upper corners.

A quantitative comparison with the Ghia benchmark results is made by plotting velocity profiles through the geometric center of the cavity—the u -velocity profile through a vertical

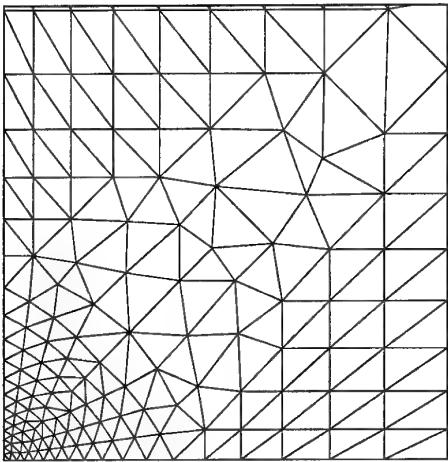


Fig. 3.4 – Mesh in the lower-left corner of the domain, $0 \leq x, y \leq 0.05$, showing the transition from wedge-shaped triangles to isotropic triangles in the corners.

cut and the v -velocity profile through a horizontal cut. The results of this comparison are shown in Figure 3.8. Note that the agreement is excellent and that no velocity or pressure oscillations are present.

3.1.4 $Re = 1000$

A particle tracer visualization of the velocity field for $Re = 1000$ is shown in Figure 3.9. This figure, as well as the velocity vectors in Figure 3.10 show the primary vortex core located very near the center of the cavity $(x, y) = (0.525, 0.572)$ at this Reynolds number. The secondary vortices in the bottom corners are much larger than at $Re = 100$. The associated pressure field is displayed via contour lines in Figure 3.11.

Again, a quantitative comparison with the Ghia benchmark results is made by plotting velocity profiles through the geometric center of the cavity—the u -velocity profile through a vertical cut and the v -velocity profile through a horizontal cut. The results of this comparison are shown in Figure 3.12. As before, note that the agreement is excellent—the mesh is stretched sufficiently toward the walls and corners to capture the velocity profile, although only one-fifth as many nodes are used.

3.1.5 $Re = 5000$

A particle tracer visualization of the velocity field for $Re = 5000$ is shown in Figure 3.13. This figure, as well as the velocity vectors in Figure 3.14, show the primary vortex core located very near the center of the cavity $(x, y) = (0.515, 0.530)$ at this Reynolds number. The secondary vortices in the bottom corners are apparent, as well as a new vortex that

develops near the upper left corner. The associated pressure field is displayed via contour lines in Figure 3.15.

Again, a quantitative comparison with the Ghia benchmark results is made by plotting velocity profiles through the geometric center of the cavity—the u -velocity profile through a vertical cut and the v -velocity profile through a horizontal cut. The results of this comparison are shown in Figure 3.16. As before, note that the agreement is excellent—the mesh is stretched sufficiently toward the walls and corners to capture the velocity profile, although only one-fifth as many nodes are used.

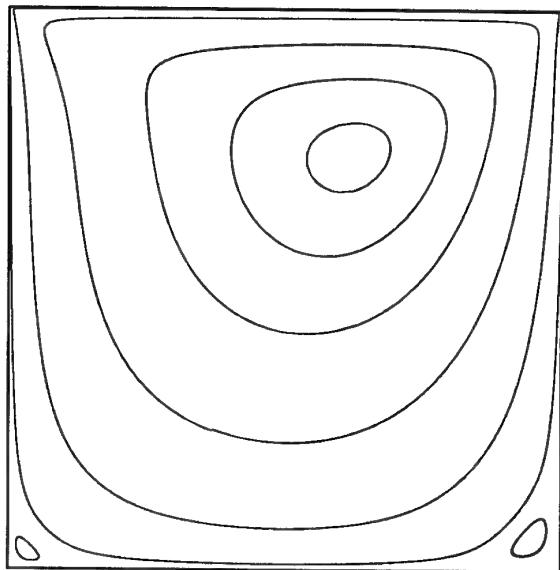


Fig. 3.5 – Lid-driven cavity flow ($Re = 100$): streamtrace visualization of the flow field.

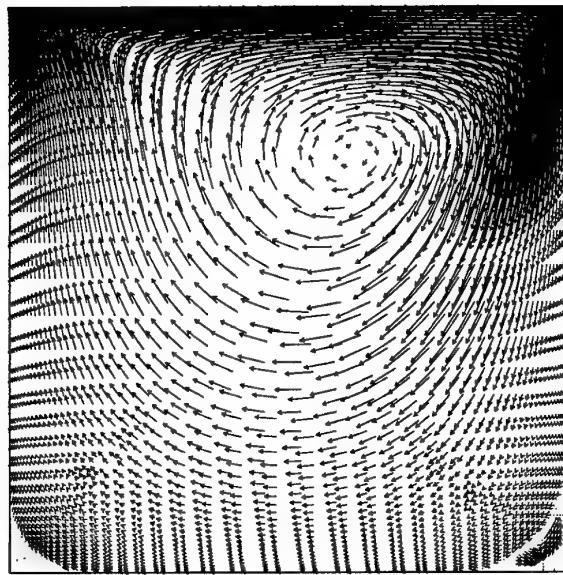


Fig. 3.6 – Lid-driven cavity flow ($Re = 100$): velocity vector visualization of the flow field.

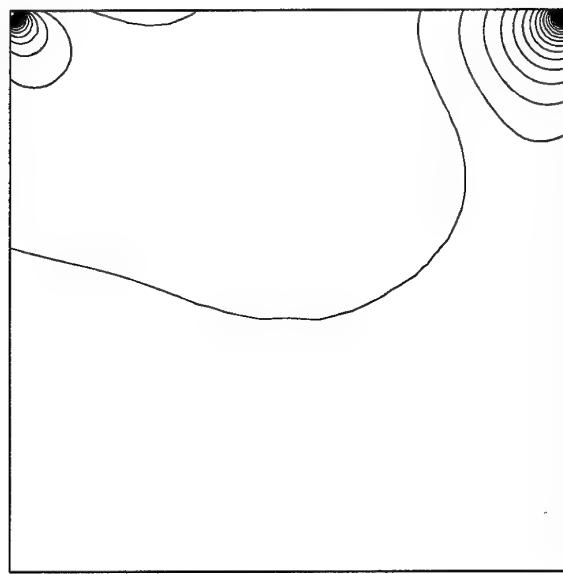


Fig. 3.7 – Lid-driven cavity flow ($Re = 100$): contour plot visualization of the pressure field.

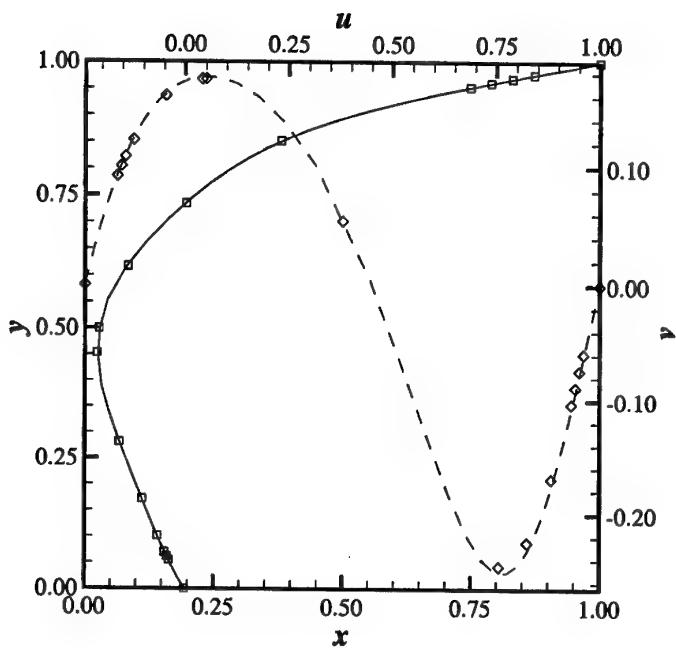


Fig. 3.8 – Lid-driven cavity flow ($Re = 100$): comparison of the results of this work (lines) with the benchmark solution of Ghia et al (Ghia et al., 1988) (markers). The $u(x = 0.5, y)$ profile is shown via the solid line and square markers. The $v(x, y = 0.5)$ profile is shown via the dashed line and diamond markers.

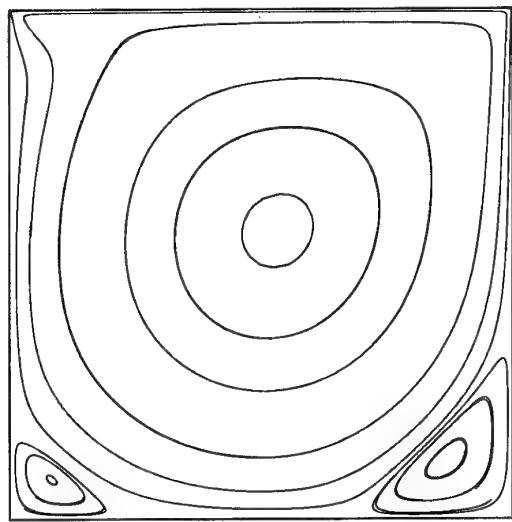


Fig. 3.9 – Lid-driven cavity flow ($Re = 1000$): streamtrace visualization of the flow field.

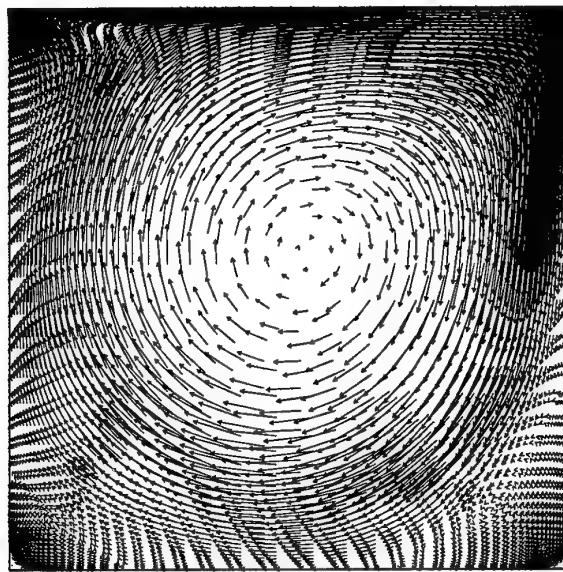


Fig. 3.10 – Lid-driven cavity flow ($Re = 1000$): velocity vector visualization of the flow field.

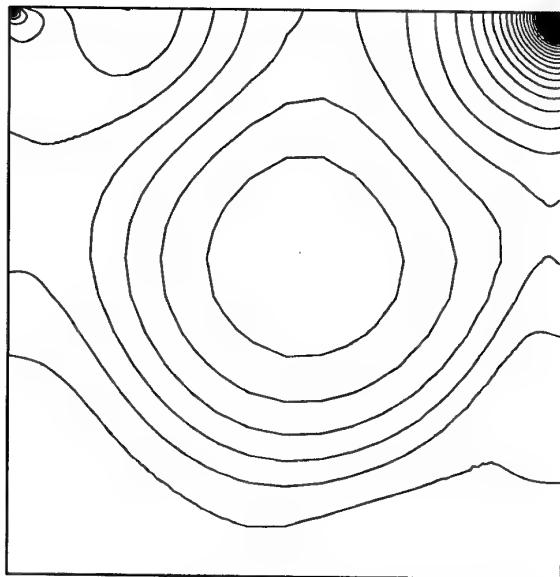


Fig. 3.11 – Lid-driven cavity flow ($Re = 1000$): contour plot visualization of the pressure field.

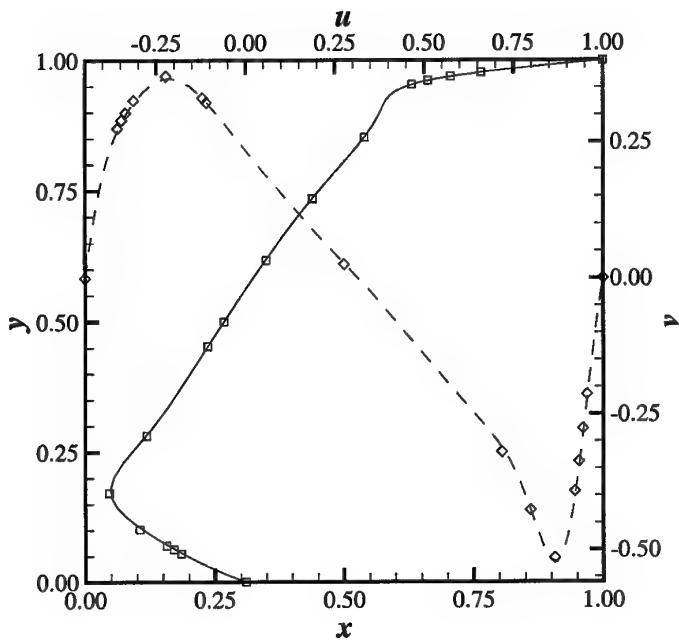


Fig. 3.12 – Lid-driven cavity flow ($Re = 1000$): comparison of the results of this work (lines) with the benchmark solution of Ghia et al (Ghia et al., 1988) (markers). The $u(x = 0.5, y)$ profile is shown via the solid line and square markers. The $v(x, y = 0.5)$ profile is shown via the dashed line and diamond markers.

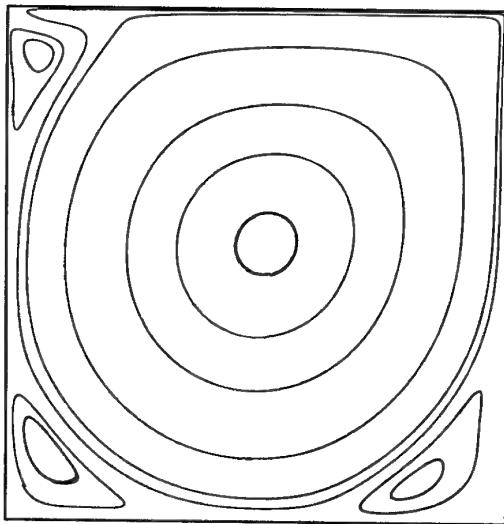


Fig. 3.13 – Lid-driven cavity flow ($Re = 5000$): streamtrace visualization of the flow field.

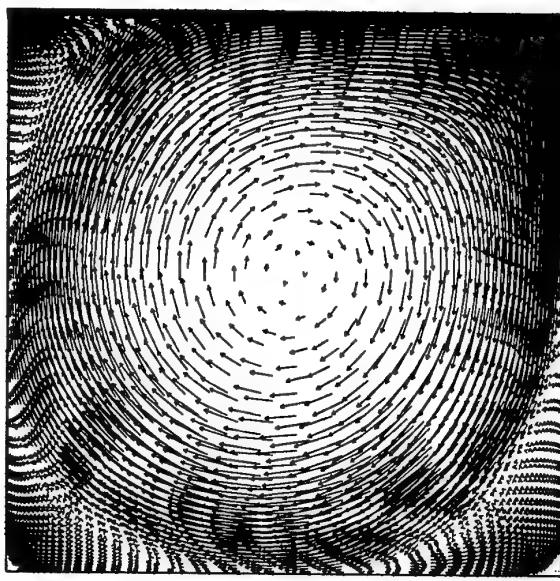


Fig. 3.14 – Lid-driven cavity flow ($Re = 5000$): velocity vector visualization of the flow field.

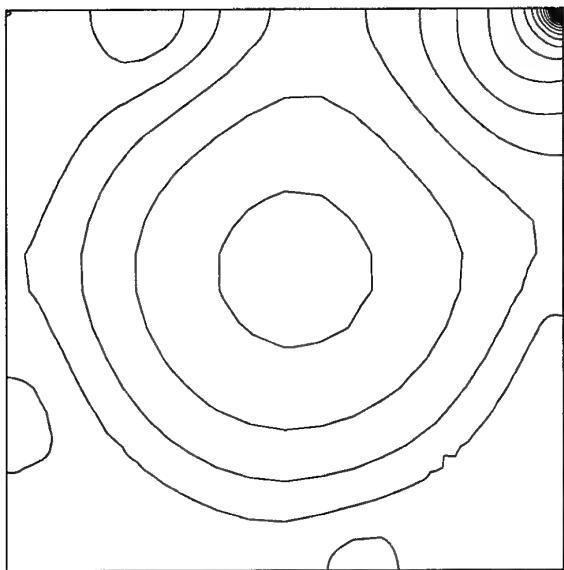


Fig. 3.15 – Lid-driven cavity flow ($Re = 5000$): contour plot visualization of the pressure field.

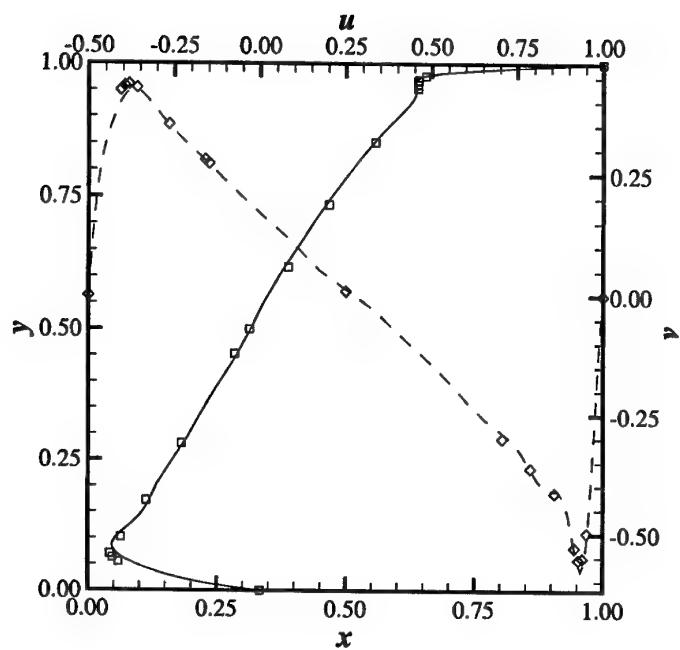


Fig. 3.16 – Lid-driven cavity flow ($Re = 5000$): comparison of the results of this work (lines) with the benchmark solution of Ghia et al (Ghia et al., 1988) (markers). The $u(x = 0.5, y)$ profile is shown via the solid line and square markers. The $v(x, y = 0.5)$ profile is shown via the dashed line and diamond markers.

3.2 Backward-Facing Step Flow

3.2.1 Problem Description

The backward-facing step case is flow in a channel having a sudden asymmetric expansion. Separated flows resulting from such geometry changes are common in industrial internal flow applications, where the performance of the device is very much dependent on the structure of the flowfield. Backward-facing step flow is also a good CFD test case because the accuracy of the solver can be tested by measuring the attachment length of the separation bubble as a function of Reynolds number and comparing with experimental values.

Backward-facing step flows are characterized by the Reynolds number and the expansion ratio (channel height downstream of the step divided by channel height upstream of the step). For this test case the geometry and flow conditions of Armaly et al (Armaly et al., 1983), shown in Figure 3.17, are simulated. This particular experimental study was chosen because great care was taken to ensure 2D flow upstream of the step and attachment lengths are presented for varying Reynolds numbers. Also, at the expansion ratio chosen by Armaly et al, a secondary recirculation cell may appear, providing yet another check on the accuracy of the solver. The flow enters through a long entrance channel (4.08 step heights (S) long in the experiment), past the step, and through the exit channel ($> 67.5S$ long in the experiment). The upstream channel height is $h = 1.06S$ and the downstream channel height is $H = 2.06S$, resulting in a channel expansion ratio of 1:1.94. Armaly et al provides results for Reynolds numbers ranging from $Re = 100$ to $Re = 7000$. Three Reynolds number values were simulated using SFE2D, all of which come from the laminar regime ($Re < 1200$): 150, 500, and 1000. The definition of the Reynolds number used in this case is given by

$$Re = \frac{VD}{\nu} \quad (3.1)$$

where V is two-thirds of the measured maximum inlet velocity (which corresponds to the average inlet velocity), D is the hydraulic diameter of the inlet channel (twice its height), and ν is the kinematic viscosity.

The flowfield contains a primary recirculation cell brought about by separation off the step edge. At certain Reynolds numbers, a secondary cell appears on the top wall due to the adverse pressure gradient created by the sudden expansion. The existence of this secondary recirculation cell is dependent on the expansion ratio and the Reynolds number. According to Armaly's experiments, at $Re = 150$ only the primary recirculation cell exists, while at $Re = 500$ and 1000 the secondary recirculation cell is also apparent.

The three calculated lengths x_1 , x_4 , and x_5 shown in Figure 3.17 are compared with the published experimental values. The length x_1 is the distance from the step edge to the primary reattachment point on the lower wall. The distance from the step edge to the separation point of the secondary recirculation cell on the top wall is x_4 , and the distance

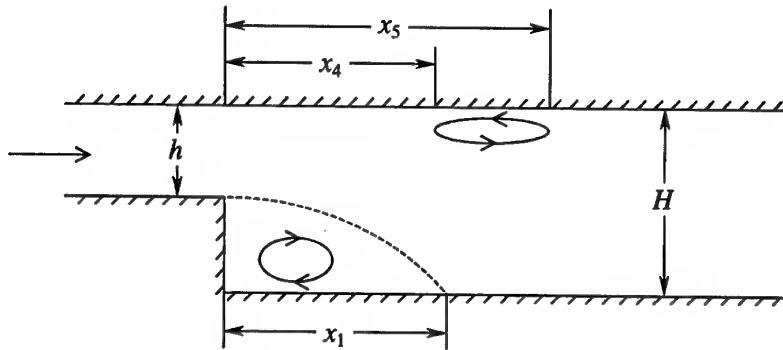


Fig. 3.17 – Flow over a backward-facing step: geometry with pertinent lengths and general flowfield properties.

from the step edge to the reattachment point of the secondary recirculation cell on the top wall is x_5 .

In comparing the results of SFE2D to Armaly's, it is important to know the experimental error in the measured data. Unfortunately, his paper gives few clues as to the uncertainty of the results. This uncertainty is compounded because the original quantitative data is not available. Instead, one is forced to read the points off the graph included in Armaly's paper. In this case, the graph was scanned into a computer at a high resolution, and the data points were digitized from the resulting image. The errors from this procedure alone are estimated at 5–8%.

3.2.2 Computational Mesh

The computational mesh used in this case consists of 12,000 nodes, which are clustered in the area of the step and down the channel $21S$, where the recirculation regions are expected. Figure 3.18 shows the mesh in the region of the step, giving an idea of the mesh density. Additionally, a mesh containing 20,000 nodes was created in order to check grid convergence of the solutions. The domain extends $5S$ upstream of the step and $35S$ downstream. Note that in this case there are no boundary layer ‘wedge’ elements. Rather, all elements are nearly isotropic triangles.

The Gambit unstructured mesh generator (Fluent, Inc.) (flu, 1998) was used to create the mesh. This generator allows for meshes of comparable quality to Delaundo (including the capability to create boundary layer ‘wedge’ elements) plus provides a graphical interface for easy mesh creation and modification.

At the inlet to the domain a fully-developed parabolic profile was prescribed of the form

$$u = \frac{6U}{h^2} ((y - y_b)h - (y - y_b)^2), \quad (3.2)$$

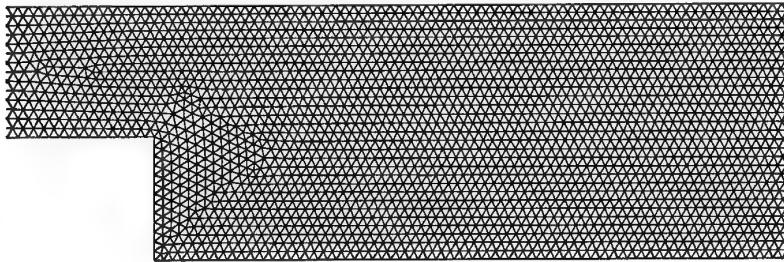


Fig. 3.18 – Flow over a backward-facing step: computational mesh in the region of the step.
The complete mesh contains approximately 12,000 nodes/24,000 elements

where U is the mean inlet velocity, h is the inlet channel height, and y_b is the y -coordinate of the inlet channel bottom wall. At the outlet, a Neumann velocity boundary condition was applied. Finally, no-slip conditions were applied on all channel walls.

3.2.3 Results

Steady-state results were obtained by starting with a stagnant field and advancing in time until the solution no longer varied in time. Figure 3.19 shows u -velocity contours for Reynolds numbers 150, 500, and 1500. Qualitative agreement with experimental observations is achieved. Only the primary recirculation region is apparent at $Re = 150$, while at $Re = 500$ and 1000, both the primary and secondary recirculation regions exist. For completeness, a representative ($Re = 500$) pressure field in the step region is provided in Figure 3.20.

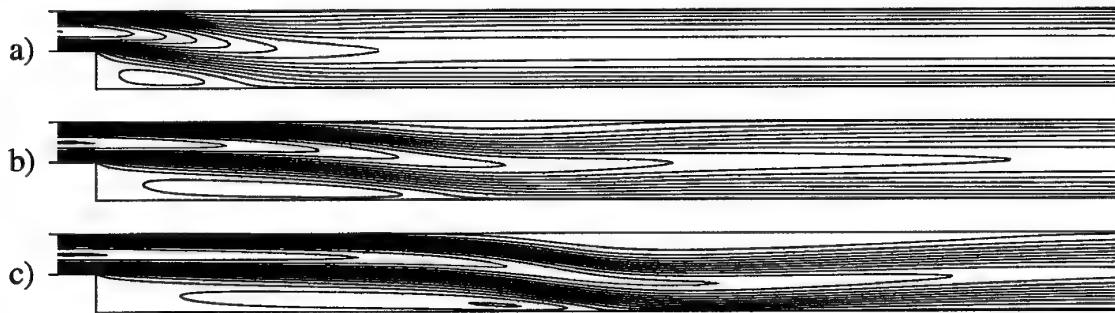


Fig. 3.19 – Flow over a backward-facing step: contours of constant streamwise velocity in the region $-1 \leq x/S \leq 27$ for Reynolds number (a) 150, (b) 500, and (c) 1000.

The computed detachment/reattachment lengths described in Figure 3.17 compared with the experimental results are shown in Table 3.1. At $Re = 150$ and 500 the errors are certainly acceptable in that, with the exception of x_5/S on the coarse mesh, they are within the uncertainty at which Armaly's data is known (5–8%). At $Re = 1000$ the errors are quite

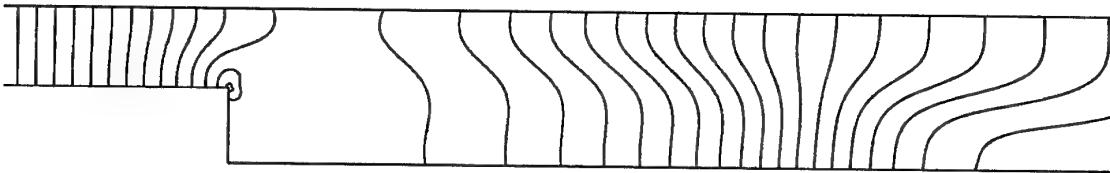


Fig. 3.20 – Flow over a backward-facing step: contours of constant pressure in the region $-3 \leq x/S \leq 8$ for Reynolds number 500.

large, however. Though larger errors are expected at higher Reynolds numbers, there may be an additional explanation. In the experiments Armaly observed that, though the flow was laminar, it was beginning to exhibit 3D features at this Reynolds number. Consequently, using a 2D solver for this flow is not strictly correct and likely adds to the error. Also shown for $Re = 1000$ are results from three commercial CFD solvers: FLOTTRAN, Flow-3D, and Fluent. These results are taken from a CFD benchmark summary compiled by Freitas (Freitas, 1995) and represent solutions provided by the respective companies for this test case. Though the grid structure, grid size, and discretization strategy may differ from those used here, it is useful to see that SFE2D provides results similar to commercial CFD codes.

Table 3.1 – Flow over a backward facing step: comparison of detachment/reattachment lengths. Experimental results are from Armaly et al (Armaly et al., 1983); numerical results compiled by Freitas (Freitas, 1995).

	x_1/S	% error	x_4/S	% error	x_5/S	% error
$Re = 150$						
Armaly et al.	4.02					
SFE2D (coarse)	3.92	2.49				
SFE2D (fine)	3.95	1.74				
$Re = 500$						
Armaly et al.	10.00		8.29		13.50	
SFE2D (coarse)	9.42	5.80	8.58	3.50	12.26	9.19
SFE2D (fine)	9.50	5.00	8.27	0.24	12.57	6.89
$Re = 1000$						
Armaly et al.	16.20		13.30		21.70	
SFE2D (coarse)	13.31	17.84	10.49	21.13	24.58	13.27
SFE2D (fine)	13.33	17.72	10.85	18.42	24.06	10.88
FLOTTRAN	8.57	47.10	6.23	53.16	16.66	23.23
Flow-3D	12.23	24.51	9.50	28.57	22.40	3.23
Fluent	13.08	19.26	10.33	22.33	24.70	13.82

Finally, Figure 3.21 shows streamwise velocity profiles at three locations in the channel

($x/S = -1.76$, 7.04, and 19.04) and $Re = 1095$. The agreement between the coarse mesh solution, fine mesh solution, and experiment is satisfactory.

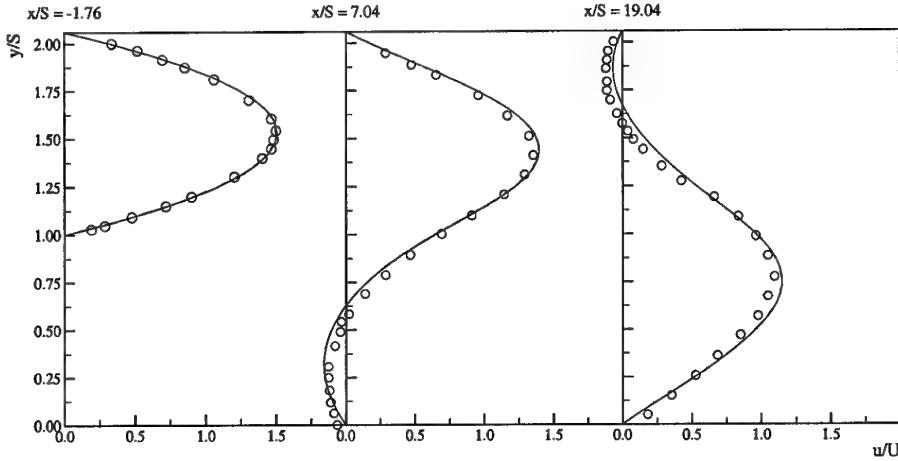


Fig. 3.21 – Flow over a backward-facing step: streamwise velocity profiles at three locations in the channel and $Re = 1095$. SFE2D coarse mesh (dashed line), SFE2D fine mesh (solid line), Armaly et al (Armaly et al., 1983) (○).

3.3 Circular Cylinder Flow

3.3.1 Problem Description

The flow over a circular cylinder is interesting from an engineering viewpoint because it contains several distinct flow characteristics, including boundary layers, flow separation, shear layers, and a wake region. This flow is certainly among the most extensively studied, both numerically and experimentally. Consequently, it is an excellent test case because a large amount of data is available for comparison. As a consequence of this vast amount of available data, review articles are a good means to gain an understanding of the basic characteristics of this flow. A number of reviews have been published from as far back as 1964, including Morvokin (Morkovin, 1964), Berger and Wille (Berger and Wille, 1972), Beaudan and Moin (Beaudan and Moin, 1994), and Williamson (Williamson, 1996b).

The behavior of flow over a circular cylinder is highly dependent upon Reynolds number; in this case typically based upon the cylinder diameter D and the freestream velocity U_0 , or

$$Re_D = \frac{U_0 D}{\nu}. \quad (3.3)$$

At very low Reynolds numbers, approximately $5 < Re_D < 40$, steady laminar flow exists with a pair of symmetric counter-rotating vortices behind the cylinder. This flow structure is

seen in Figure 3.22(a) taken from VanDyke (VanDyke, 1982), which shows an experimental visualization at $Re_D = 26$. The range of Reynolds numbers from approximately $40 < Re_D < 190$ is often termed the laminar vortex shedding regime. The recirculation cells develop instabilities, whose strength and amplification grow with Re_D . These instabilities induce laminar vortex shedding, also known as the Karman vortex street. Figure 3.22(b), also taken from VanDyke (VanDyke, 1982) shows an experimental visualization of the Karman vortex street at $Re_D = 140$.

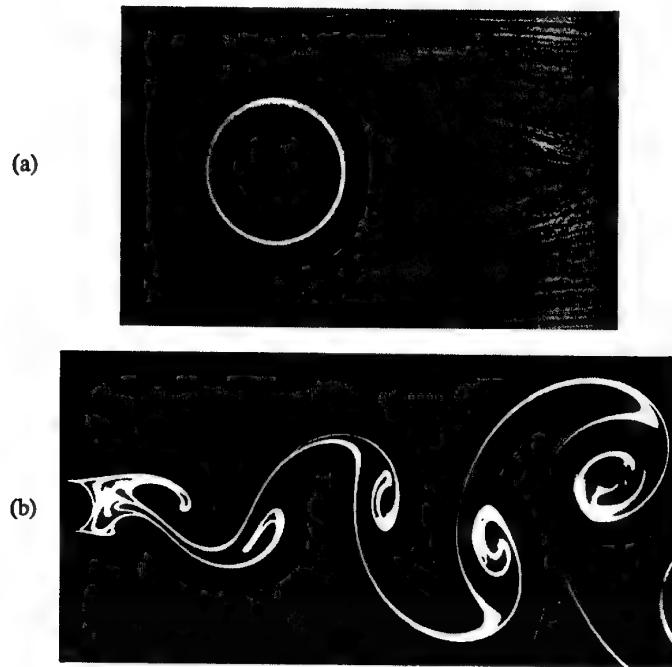


Fig. 3.22 – 2D circular cylinder flow: experimental flow visualizations from VanDyke (VanDyke, 1982). (a) $Re_D = 24$, (b) $Re_D = 140$.

At Reynolds numbers above approximately 190, the wake behind the cylinder becomes unstable to transverse perturbations and develops 3D structures. As a consequence, 2D calculations tend to yield incorrect values of the flow parameters (lift, drag, shedding frequency, etc.) at these Reynolds numbers. For this 2D code validation then, the maximum Reynolds number used was 140 ... well below the initiation of 3D flow features.

A number of flow quantities can be computed and compared with existing numerical and experimental data. In this study, the pressure drag,

$$C_{DP} = \frac{1}{\frac{1}{2}\rho U_0^2 D} \int_0^{2\pi} P \cos \theta \, d\theta, \quad (3.4)$$

viscous drag,

$$C_{D_v} = \frac{1}{\frac{1}{2}\rho Re_D U_0^3 D} \int_0^{2\pi} \omega_w \sin \theta \, d\theta, \quad (3.5)$$

and total drag

$$C_D = C_{D_P} + C_{D_v}, \quad (3.6)$$

were validated. In addition, at the unsteady Reynolds numbers the non-dimensional vortex shedding frequency, or Strouhal number,

$$St = \frac{fD}{U_0}, \quad (3.7)$$

was also validated. In the above equations, ρ is the density, U_0 the freestream velocity, D the cylinder diameter, f the shedding frequency, ω_w the wall vorticity, and P the pressure.

3.3.2 Computational Mesh

The computational domain, displayed in Figure 3.23, is rectangular in shape and extends $20D$ upstream, $20D$ downstream, and $30D$ laterally from the center of the cylinder. Kravchenko and Moin (Kravchenko and Moin, 1998) found that C_D , C_L , and St do not become independent of lateral domain size until approximately $60D$. However, the relative difference between these quantities is less than 4% when the lateral boundaries are moved from $120D$ to $30D$.

The domain was discretized into 31,000 linear triangle elements with 16,000 nodes at the vertices. As seen in Figure 3.24, elements are clustered in the wake region, with the characteristic element side length being approximately $0.25D$. Outside the wake region the elements are less dense, with characteristic element side lengths of $0.5D$ at the upstream and lateral extents. Figure 3.24 shows the spatial discretization near the cylinder. Elements are clustered toward the cylinder in order to better capture the initial development of the shed vorticity, with element side lengths on the cylinder wall of $0.028D$. Boundary layer wedge-type elements were used very near the cylinder walls to capture the boundary layer, with the perpendicular dimension of the first element off the cylinder being $0.005D$.

The boundary conditions were as follows. At the upstream boundary, a fixed-velocity inlet condition was used, with $u = U_0$ and $v = 0$. A standard Neumann velocity outlet condition was utilized at the downstream boundary, while slip wall conditions were used at the lateral boundaries. On the cylinder surface a no-slip wall condition was used.

The time step was $\Delta t U_0 / D = 0.01$, resulting in approximately 550 time steps per oscillation period of the lift force for $Re_D = 140$ —slightly more for the $Re_D = 100$ and 60 cases.

Finally, mesh convergence was verified by running fine-mesh simulations. The mesh used for these simulations had characteristic element side lengths of half that of the coarse mesh

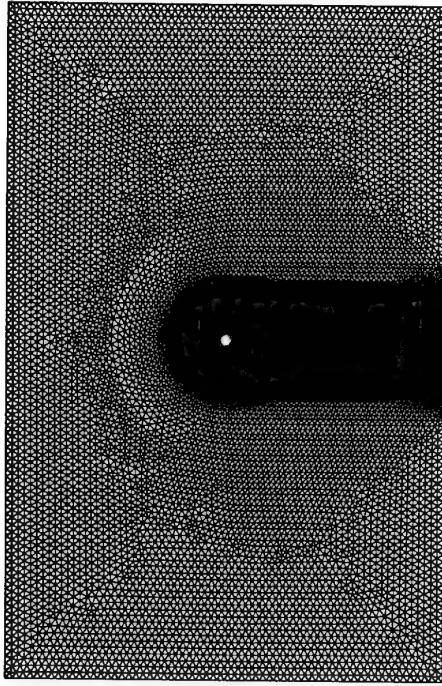


Fig. 3.23 – 2D circular cylinder flow: overview of the computational domain and mesh.

in the wake region, along the cylinder wall, and in the boundary layer. This mesh refinement resulted in a total of 79,000 elements with 40,000 nodes. In addition, the time step was halved to $\Delta t U_0 / D = 0.005$ for these fine-mesh simulations.

3.3.3 Results

Simulations were performed at one steady ($Re_D = 20$) and three unsteady ($Re_D = 60, 100$, and 140) Reynolds numbers. In each case, potential flow was used as the initial condition and the simulation advanced in time until a statistically stationary flow pattern was developed. In the initial simulations, the vortex shedding was allowed to develop without introducing perturbations to the flow. That is, the shedding was induced by numerical truncation and round-off errors that eventually break the symmetry of the solution. This turned out to be a very slow process. So, in order to initiate vortex shedding more quickly, a time-dependent rotation was applied to the cylinder for a short time. The flow was then advanced in time until all transient features exited the domain.

The qualitative behavior of the computed flowfield is seen via contours of constant vorticity shown in Figure 3.25. At $Re_D = 20$, two stationary attached vortices are formed

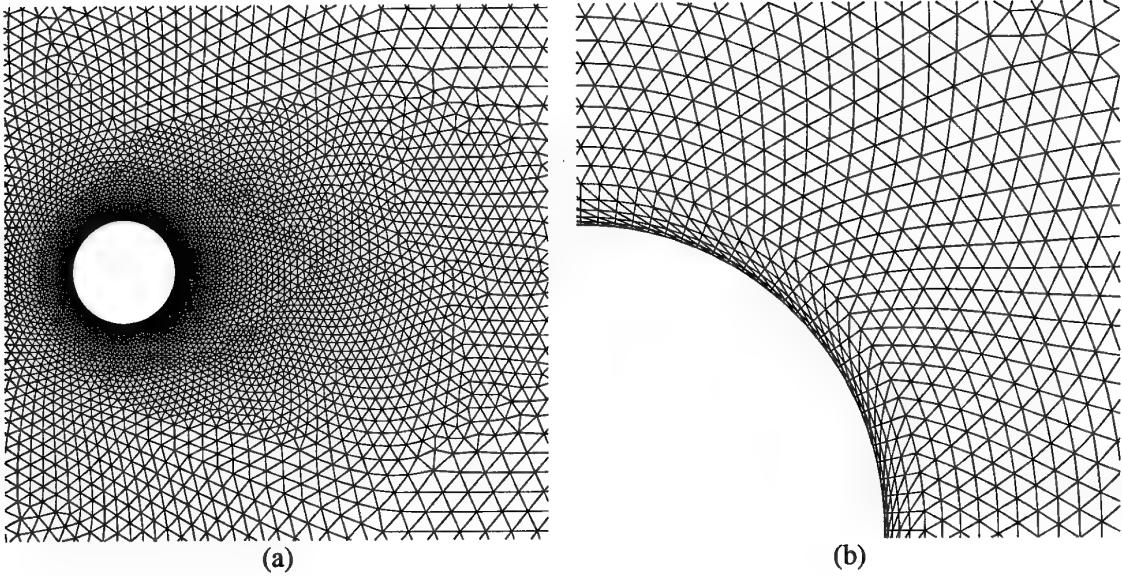


Fig. 3.24 – 2D circular cylinder flow: the computational mesh near the cylinder. (a) shows the clustering of elements toward the cylinder, while (b) shows the boundary layer wedge-type elements very near the cylinder wall.

behind the cylinder. At the higher Reynolds numbers, the wake behind the cylinder consists of negative and positive vortices shed alternately from the upper and lower portions of the cylinder surface.

Quantitative validation of the flow solver is achieved by comparing C_{D_P} , C_{D_v} , and St with published values. The drag coefficient benchmarks are taken from Henderson (Henderson, 1995). Henderson's curves are fits to numerical data from high-order 2D spectral-element simulations that produced results in excellent agreement with the experiments of Williamson and Roshko (Williamson and Roshko, 1990). The Strouhal number benchmark is the so-called “universal” Strouhal curve for a circular cylinder as put forth by Williamson (Williamson, 1988). This curve is a fit through numerous experiments conducted using various techniques.

The SFE2D results compared with the benchmark values are shown in Figure 3.26. The agreement is excellent for all Reynolds numbers. Also seen in this figure is that coarse- and fine-mesh results are virtually indistinguishable from one another. Consequently, the solution is deemed grid converged.

As a final note, the SFE2D solver results can be compared with the results of commercial CFD codes. The $Re_D = 60$ case is one of a number of test cases included in the CFD benchmark summary compiled by Freitas (Freitas, 1995). The comparison is summarized in Table 3.2. Based on the experimental work of Tritton (Tritton, 1959), the measured

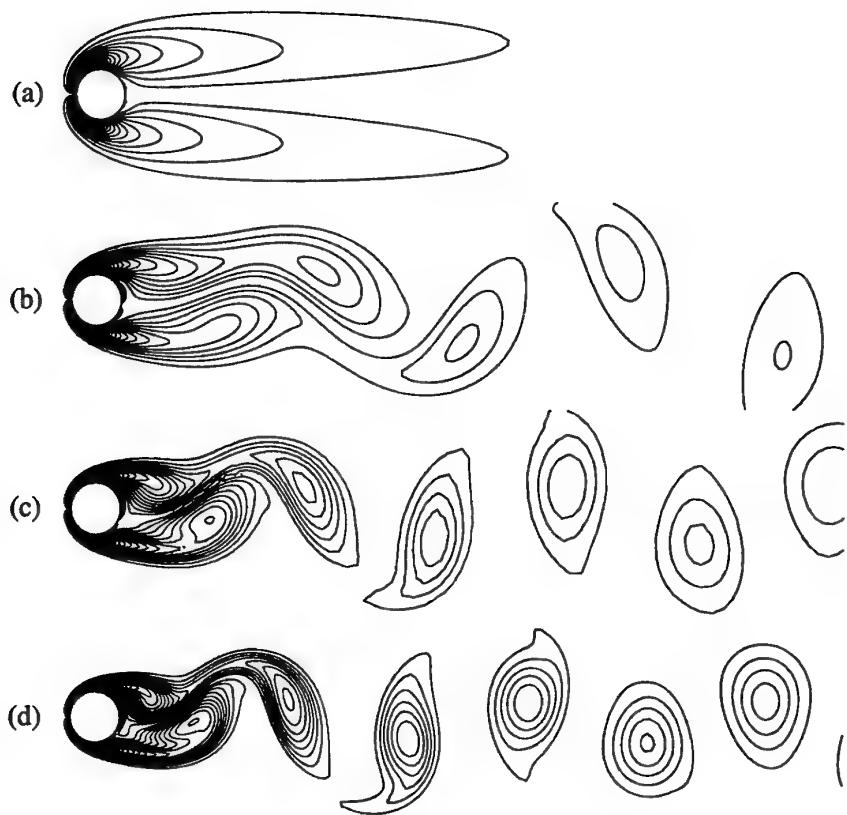


Fig. 3.25 – 2D circular cylinder flow: 50 instantaneous contours of constant vorticity magnitude ($0 \leq |\omega|D/U_0 \leq 150$) in the region $-1.0D \leq x \leq 15.0D$, $-2.0D \leq y \leq 2.0D$. (a) $Re_D = 20$, (b) $Re_D = 60$, (c) $Re_D = 100$, (d) $Re_D = 140$. Flow patterns (a)-(d) correspond to the maximum value of the lift coefficient.

drag coefficient is 1.47. According to Schlichting (Schlichting, 1979), the Strouhal number is measured to be 0.14. Though the grid structure, grid size, and discretization strategy may differ, it is useful to see that SFE2D provides results at least as accurate as the commercial CFD codes.

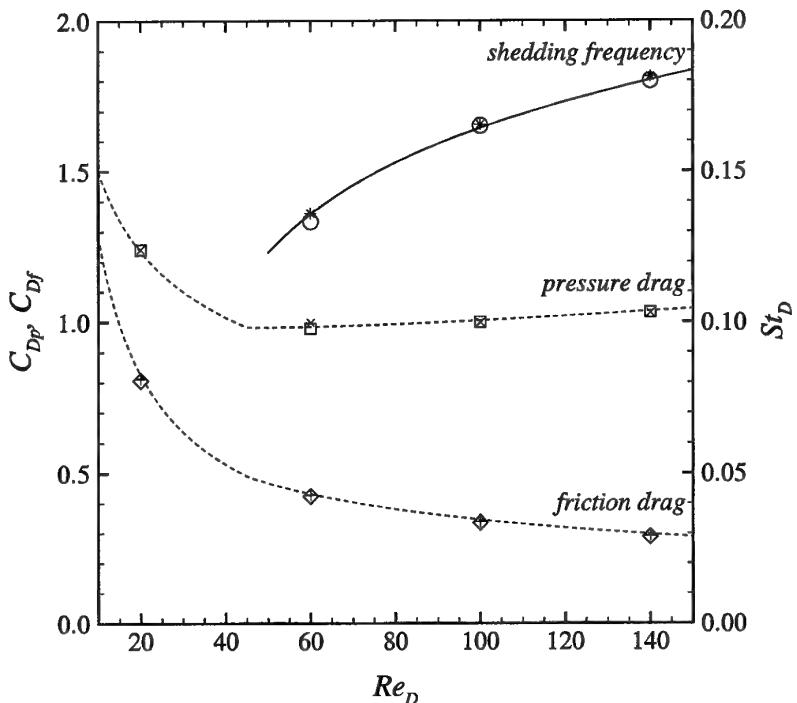


Fig. 3.26 – 2D circular cylinder flow: quantitative comparison with published values of C_{D_p} , C_{D_v} , and St vs. Reynolds number. —, experimental fit by Williamson (Williamson, 1988); - - -, numerical fit by Henderson (Henderson, 1995); \circ , \diamond , \square , SFE2D coarse mesh; $*$, $+$, \times , SFE2D fine mesh.

Table 3.2 – 2D circular cylinder flow: comparison of results from commercial CFD solvers (Freitas, 1995) for drag coefficient C_D and Strouhal Number St at $Re = 60$. The experimental values for C_D and St are taken from Tritton (Tritton, 1959) and Schlichting (Schlichting, 1979), respectively.

	Lateral/inlet/outlet distance	Grid resolution	C_D	St
Experiment			1.47	0.14
SFE2D	30D/20D/20D	16,000 nodes (uns)	1.405	0.137
FLOW-3D	6.5D/2.5D/22D	198 × 80 (cartesian)	1.77	0.15
FLOTTRAN	8D/~10D/~20D	21,498 nodes (uns)	1.44	NA
FLUENT	5D/NA/20D	101 × 201 (O-H)	1.567	0.15
CFDS-FLOW3D	NA/3.5D/9.5D	3,384 nodes (block-str)	0.037	NA
NISA/3D-FLUID	10D/8D/20D	3,603 nodes (uns)	1.343	NA

4. 3D LAMINAR FLOW SOLVER

In this chapter, extension of the 2D stabilized FE solver derived in Chapter 2 to a 3D FE/spectral solver is explained. Taking inspiration from Henderson (Henderson, 1999), this solver utilizes an in-plane SUPG/PSPG FEM discretization and a collocated spectral discretization in the transverse direction to recast the 3D problem into a set of coupled 2D problems in Fourier space.

First, the governing equations are presented in a form that separates the in-plane and transverse components. The discretization of these equations, first in the plane and then in the transverse direction, is explained. A description of the special treatment of the convective terms (including SUPG terms) required to reduce computation cost follows. The parallelization approach is discussed and, finally, the details of the solver implementation are presented by means of an in-depth algorithm description.

4.1 Governing Equations

Consider laminar, incompressible, isothermal flow in the absence of body forces. The governing Navier-Stokes equations are

$$\frac{\partial \vec{u}}{\partial t} + (\vec{u} \cdot \nabla) \vec{u} = -\nabla p + \nu \nabla^2 \vec{u} \quad (4.1)$$

$$\nabla \cdot \vec{u} = 0, \quad (4.2)$$

where \vec{u} is the 3D velocity vector, p is the kinematic pressure (pressure divided by density) and ν is the kinematic viscosity.

The solution technique developed in this work employs a different discretization method for the in-plane and transverse directions. Consequently, the governing equations are now written in a form that separates these components. In order to accomplish this, the following notation is adopted for the remainder of this work.

(x, y)	in-plane coordinates
z	transverse coordinate
\vec{u}	2D velocity vector, (u, v)
w	transverse velocity component
$\tilde{\nabla}$	2D gradient operator, $(\partial/\partial x, \partial/\partial y)$

Equations (4.1) and (4.2) can now be written as

$$\frac{\partial \vec{u}}{\partial t} + (\vec{u} \cdot \tilde{\nabla}) \vec{u} + w \frac{\partial \vec{u}}{\partial z} = -\tilde{\nabla} p + \nu \left(\tilde{\nabla}^2 + \frac{\partial^2}{\partial z^2} \right) \vec{u} \quad (4.3)$$

$$\frac{\partial w}{\partial t} + (\vec{u} \cdot \tilde{\nabla}) w + w \frac{\partial w}{\partial z} = -\frac{\partial p}{\partial z} + \nu \left(\tilde{\nabla}^2 + \frac{\partial^2}{\partial z^2} \right) w \quad (4.4)$$

$$\tilde{\nabla} \cdot \vec{u} + \frac{\partial w}{\partial z} = 0. \quad (4.5)$$

4.2 Temporal Discretization

The temporal discretization of the 3D solver is identical to that of the 2D solver. For the pressure and diffusion terms, a second-order Crank-Nicolson method is applied. In terms of a simple example first-order ordinary differential equation (ODE)

$$\frac{df}{dt} = \lambda f, \quad (4.6)$$

this method results in the discrete expression

$$\frac{f^{n+1} - f^n}{\Delta t} = \lambda \frac{f^{n+1} + f^n}{2}, \quad (4.7)$$

where $[]^{n+1}$ denotes the new time step and $[]^n$ the current time step.

It is apparent from (4.7) that the Crank-Nicolson method is based on central differencing (centered about time $n + \frac{1}{2}$) and hence is second-order accurate. Due to the implicitness of the scheme, it is mathematically deemed unconditionally stable for all time step values, meaning that perturbations are not amplified with time. In practice, though, it is found that oscillations can occur for sufficiently large time step values.

The convective terms are treated using a second-order explicit Adams-Bashforth method. Again, in terms of the example first-order ODE (4.6), the method results in the discrete expression

$$\frac{f^{n+1} - f^n}{\Delta t} = \lambda \left(\frac{3}{2} f^n - \frac{1}{2} f^{n-1} \right). \quad (4.8)$$

As with all explicit schemes, the Adams-Bashforth method imposes a stability limit on the time step size. As mentioned in Chapter 2, although this can be a serious limitation in some cases, an explicit treatment of the convection terms is acceptable (and even desirable) in this work for three primary reasons.

- The convection terms are non-linear. Therefore, if treated explicitly, the system is linear within each time step and there is no need for multiple iterations within a time step or the necessity to calculate expensive Jacobian matrices.
- The ultimate intent of this solver is to perform LES. Since LES requires very small time steps anyway, the time step limitation due to the explicit convection treatment is less important.

- As will be shown later, explicit treatment of the non-linear terms decouples the Fourier modes within each time step when the FE/spectral discretization is used.

4.3 Spectral Methods

Spectral methods are characterized by the expansion of the solution in terms of global, orthogonal polynomials. This global expansion, as opposed to local expansions used in the FDM, FVM, and FEM, allows a much faster (exponential) convergence rate, which is the main advantage of the method.

Numerical spectral methods for PDEs were originally developed by meteorologists some half a century ago (Silberman, 1954). However, nonlinearities present a considerable computational expense, which was a severe drawback to the method until alternative “pseudospectral” transform methods (to be discussed shortly) were developed (Orzag, 1969; Eliassen et al., 1970).

The main difficulty in applying spectral methods to general fluid dynamics problems arises in implementing boundary conditions on anything other than simple geometries. To work around this problem, spectral multi-domain (or spectral element) techniques have been developed and remain a current area of intense research (Patera, 1984; Henderson, 1999). In these methods, the full, complex domain is partitioned into subdomains of simple geometry and a spectral method applied in each subdomain.

A number of different spectral methods have been developed. The reader is referred to Canuto et al. (Canuto et al., 1988) or Hussaini and Zang (Hussaini and Zang, 1987) for a review of spectral methods and their application to fluid dynamics. In this work, the flow is assumed to be periodic in the transverse direction. Therefore, a simple collocated Fourier spectral method, discussed later in this chapter, is well suited.

4.4 Finite-Element/Spectral Discretization

Two options exist in performing the spatial discretization of equations (4.3)–(4.5). One can first perform the Fourier decomposition in the periodic direction and then the FEM discretization in the plane, or vice-versa. For a pure Galerkin or Galerkin/PSPG formulation, the two approaches produce the same result. However, the use of convective stabilization (SUPG) requires that the in-plane discretization be performed first, and then the Fourier decomposition in the transverse direction.

The reason for this can immediately be seen if the Fourier decomposition is performed first. To do so, we assume the flow to be periodic in the transverse (z) direction and known at discrete points in space. Consequently, the flowfield can be developed as a discrete Fourier

sum. That is, any quantity $q_n(x, y, t) = q(x, y, z_n, t)$ is expressed as

$$q_n(x, y, t) = \frac{1}{N} \sum_{k=0}^{N-1} \hat{q}_k(x, y, t) e^{\frac{2\pi I k}{N} z_n} \quad (4.9)$$

or, equivalently

$$q(x, y, z_n, t) = \frac{1}{N} \sum_{k=0}^{N-1} \hat{q}_k(x, y, f_k, t) e^{\frac{2\pi I k}{L} z_n} \quad (4.10)$$

where $I \equiv \sqrt{-1}$, N is the number of discrete samples in the set, and L is the dimension of the transverse domain. The equations for the discrete Fourier modes \hat{q}_k are obtained by taking the discrete Fourier transform (DFT) of the governing equations (4.3)–(4.5). The DFT is defined as

$$\hat{q}_k(x, y, t) = \sum_{n=0}^{N-1} q_n(x, y, t) e^{-\frac{2\pi I n}{N} k}. \quad (4.11)$$

Upon execution of the transform, the governing equations for the Fourier coefficients of the velocity vector and pressure are

$$\frac{\partial \vec{u}_k}{\partial t} + \underbrace{\sum_{n=0}^{N-1} \left[(\vec{u}_n \cdot \tilde{\nabla}) \vec{u}_n + w_n \frac{\partial \vec{u}_n}{\partial z} \right] e^{-\frac{2\pi I n}{N} k}}_{\vec{h}_k} = -\tilde{\nabla} \hat{p}_k + \nu \left(\tilde{\nabla}^2 - (2\pi k/L)^2 \right) \vec{u}_k \quad (4.12)$$

$$\frac{\partial \hat{w}_k}{\partial t} + \underbrace{\sum_{n=0}^{N-1} \left[(\vec{u}_n \cdot \tilde{\nabla}) w_n + w_n \frac{\partial w_n}{\partial z} \right] e^{-\frac{2\pi I n}{N} k}}_{\hat{h}_{z k}} = -\tilde{\nabla} \hat{p}_k + \nu \left(\tilde{\nabla}^2 - (2\pi k/L)^2 \right) \vec{u}_k \quad (4.13)$$

$$\tilde{\nabla} \cdot \vec{u} + (2\pi I k/L) \hat{w}_k = 0. \quad (4.14)$$

Note that the convective terms \vec{h}_k and $\hat{h}_{z k}$ in the above expressions have not yet been evaluated. Now, because the convective terms are nonlinear, their Fourier transforms cannot be written in a form $(\vec{u}_k \cdot \tilde{\nabla}) \vec{u}_k$ and $(\vec{u}_k \cdot \tilde{\nabla}) \hat{w}_k$ necessary to be able to define the SUPG terms as described in Section 2.2.2.

So, one must first define a SUPG/PSPG semi-discretization for the original 3D equations

(4.3)–(4.5). Following the procedure given in Chapter 2, we derive

$$\sum_e \left[\int_{\Omega_e} N_i \left(\frac{\partial u}{\partial t} + (\vec{u} \cdot \tilde{\nabla}) u + w \frac{\partial u}{\partial z} - \nu \frac{\partial^2 u}{\partial z^2} \right) d\Omega_e - \right. \\ \left. \int_{\Omega_e} \left(\frac{\partial N_i}{\partial x} p \right) d\Omega_e + \int_{\Omega_e} (\tilde{\nabla} N_i \cdot \nu \tilde{\nabla} u) d\Omega_e + \right. \\ \left. \underbrace{\tau_{SUPG} \int_{\Omega_e} (\vec{u} \cdot \tilde{\nabla} N_i) \left(\frac{\partial u}{\partial t} + (\vec{u} \cdot \tilde{\nabla}) u + w \frac{\partial u}{\partial z} + \frac{\partial p}{\partial x} - \nu \left(\underline{\tilde{\nabla}^2 u} + \frac{\partial^2 u}{\partial z^2} \right) \right) d\Omega_e}_{SUPG} \right] = 0 \quad (4.15)$$

$$\sum_e \left[\int_{\Omega_e} N_i \left(\frac{\partial v}{\partial t} + (\vec{u} \cdot \tilde{\nabla}) v + w \frac{\partial v}{\partial z} - \nu \frac{\partial^2 v}{\partial z^2} \right) d\Omega_e - \right. \\ \left. \int_{\Omega_e} \left(\frac{\partial N_i}{\partial y} p \right) d\Omega_e + \int_{\Omega_e} (\tilde{\nabla} N_i \cdot \nu \tilde{\nabla} v) d\Omega_e + \right. \\ \left. \underbrace{\tau_{SUPG} \int_{\Omega_e} (\vec{u} \cdot \tilde{\nabla} N_i) \left(\frac{\partial v}{\partial t} + (\vec{u} \cdot \tilde{\nabla}) v + w \frac{\partial v}{\partial z} + \frac{\partial p}{\partial y} - \nu \left(\underline{\tilde{\nabla}^2 v} + \frac{\partial^2 v}{\partial z^2} \right) \right) d\Omega_e}_{SUPG} \right] = 0 \quad (4.16)$$

$$\sum_e \left[\int_{\Omega_e} N_i \left(\frac{\partial w}{\partial t} + (\vec{u} \cdot \tilde{\nabla}) w + w \frac{\partial w}{\partial z} + \frac{\partial p}{\partial z} - \nu \frac{\partial^2 w}{\partial z^2} \right) d\Omega_e + \right. \\ \left. \int_{\Omega_e} (\tilde{\nabla} N_i \cdot \nu \tilde{\nabla} w) d\Omega_e + \right. \\ \left. \underbrace{\tau_{SUPG} \int_{\Omega_e} (\vec{u} \cdot \tilde{\nabla} N_i) \left(\frac{\partial w}{\partial t} + (\vec{u} \cdot \tilde{\nabla}) w + w \frac{\partial w}{\partial z} + \frac{\partial p}{\partial z} - \nu \left(\underline{\tilde{\nabla}^2 w} + \frac{\partial^2 w}{\partial z^2} \right) \right) d\Omega_e}_{SUPG} \right] = 0 \quad (4.17)$$

$$\sum_e \left[\int_{\Omega_e} N_i \left(\tilde{\nabla} \cdot \vec{u} + \frac{\partial w}{\partial z} \right) d\Omega_e + \right. \\ \left. \underbrace{\tau_{PSPG} \int_{\Omega_e} \tilde{\nabla} N_i \cdot \left(\frac{\partial \vec{u}}{\partial t} + (\vec{u} \cdot \tilde{\nabla}) \vec{u} + w \frac{\partial \vec{u}}{\partial z} + \tilde{\nabla} p - \nu \left(\underline{\tilde{\nabla}^2 \vec{u}} + \frac{\partial^2 \vec{u}}{\partial z^2} \right) \right) d\Omega_e}_{PSPG} \right] = 0 \quad (4.18)$$

where the SUPG and PSPG stabilization terms have been identified via an underbrace. The underlined terms in the above expressions vanish for a P1 element approximation.

Now a Fourier decomposition in the z -direction is performed at the discrete level (equations (4.15)–(4.18)). After performing the DFT, one obtains the following:

$$\sum_e \left[\int_{\Omega_e} N_i \frac{\partial \hat{u}_k}{\partial t} d\Omega_e + (\hat{h}_{xk})_i - \int_{\Omega_e} \frac{\partial N_i}{\partial x} \hat{p}_k d\Omega_e + \right. \\ \left. \int_{\Omega_e} \nu \left(\tilde{\nabla} N_i \cdot \tilde{\nabla} \hat{u}_k + (2\pi k/L)^2 N_i \hat{u}_k \right) d\Omega_e + (\hat{ST}_{SUPG \times k})_i \right] = 0 \quad (4.19)$$

$$\sum_e \left[\int_{\Omega_e} N_i \frac{\partial \hat{v}_k}{\partial t} d\Omega_e + (\hat{h}_{y k})_i - \int_{\Omega_e} \frac{\partial N_i}{\partial y} \hat{p}_k d\Omega_e + \int_{\Omega_e} \nu \left(\tilde{\nabla} N_i \cdot \tilde{\nabla} \hat{v}_k + (2\pi k/L)^2 N_i \hat{v}_k \right) d\Omega_e + (\hat{S}T_{SUPG y k})_i \right] = 0 \quad (4.20)$$

$$\sum_e \left[\int_{\Omega_e} N_i \frac{\partial \hat{w}_k}{\partial t} d\Omega_e + (\hat{h}_{z k})_i + \frac{2\pi I k}{L} \int_{\Omega_e} N_i \hat{p}_k d\Omega_e + \int_{\Omega_e} \nu \left(\tilde{\nabla} N_i \cdot \tilde{\nabla} \hat{w}_k + (2\pi k/L)^2 N_i \hat{w}_k \right) d\Omega_e + (\hat{S}T_{SUPG z k})_i \right] = 0 \quad (4.21)$$

$$\sum_e \left[\int_{\Omega_e} N_i (\tilde{\nabla} \cdot \vec{u}_k) d\Omega_e + \frac{2\pi I k}{L} \int_{\Omega_e} N_i \hat{w}_k d\Omega_e + \tau_{PSPG} \int_{\Omega_e} \tilde{\nabla} N_i \cdot \left(\frac{\partial \vec{u}_k}{\partial t} + \tilde{\nabla} \hat{p}_k + \nu (2\pi k/L)^2 \vec{u}_k \right) d\Omega_e + (\hat{S}T_{PSPG C k})_i \right] = 0 \quad (4.22)$$

where the convective terms are

$$(\hat{h}_{x k})_i = \sum_{n=0}^{N-1} \int_{\Omega_e} N_i \left((\vec{u}_n \cdot \tilde{\nabla}) u_n + w_n \left(\frac{\partial u}{\partial z} \right)_n \right) e^{-\frac{2\pi I n}{N} k} d\Omega_e \quad (4.23)$$

$$(\hat{h}_{y k})_i = \sum_{n=0}^{N-1} \int_{\Omega_e} N_i \left((\vec{u}_n \cdot \tilde{\nabla}) v_n + w_n \left(\frac{\partial v}{\partial z} \right)_n \right) e^{-\frac{2\pi I n}{N} k} d\Omega_e \quad (4.24)$$

$$(\hat{h}_{z k})_i = \sum_{n=0}^{N-1} \int_{\Omega_e} N_i \left((\vec{u}_n \cdot \tilde{\nabla}) w_n + w_n \left(\frac{\partial w}{\partial z} \right)_n \right) e^{-\frac{2\pi I n}{N} k} d\Omega_e. \quad (4.25)$$

The PSPG stabilization of the convective terms is

$$(\hat{S}T_{PSPG C k})_i = \sum_{n=0}^{N-1} \tau_{PSPG} \int_{\Omega_e} \tilde{\nabla} N_i \cdot \left((\vec{u}_n \cdot \tilde{\nabla}) \vec{u}_n + w_n \left(\frac{\partial \vec{u}}{\partial z} \right)_n \right) e^{-\frac{2\pi I n}{N} k} d\Omega_e, \quad (4.26)$$

and the SUPG convective stabilization terms are

$$(\hat{S}T_{SUPG x k})_i = \sum_{n=0}^{N-1} \left\{ \tau_{SUPG} \int_{\Omega_e} \vec{u}_n \cdot \tilde{\nabla} N_i \left(\frac{\partial u_n}{\partial t} + (\vec{u}_n \cdot \tilde{\nabla}) u_n + w_n \left(\frac{\partial u}{\partial z} \right)_n + \frac{\partial p_n}{\partial x} - \nu \left(\frac{\partial^2 u}{\partial z^2} \right)_n \right) d\Omega_e \right\} e^{-\frac{2\pi I n}{N} k} \quad (4.27)$$

$$(\hat{S}T_{SUPG y k})_i = \sum_{n=0}^{N-1} \left\{ \tau_{SUPG} \int_{\Omega_e} \vec{u}_n \cdot \tilde{\nabla} N_i \left(\frac{\partial v_n}{\partial t} + (\vec{u}_n \cdot \tilde{\nabla}) v_n + w_n \left(\frac{\partial v}{\partial z} \right)_n + \frac{\partial p_n}{\partial y} - \nu \left(\frac{\partial^2 v}{\partial z^2} \right)_n \right) d\Omega_e \right\} e^{-\frac{2\pi I n}{N} k} \quad (4.28)$$

$$(\hat{S}T_{SUPG} z_k)_i = \sum_{n=0}^{N-1} \left\{ \tau_{SUPG} \int_{\Omega_e} \vec{u}_n \cdot \tilde{\nabla} N_i \left(\frac{\partial w_n}{\partial t} + (\vec{u}_n \cdot \tilde{\nabla}) w_n + w_n \left(\frac{\partial w}{\partial z} \right)_n + \left(\frac{\partial p}{\partial z} \right)_n - \nu \left(\frac{\partial^2 w}{\partial z^2} \right)_n \right) d\Omega_e \right\} e^{-\frac{2\pi I n}{N} k}. \quad (4.29)$$

As before, the evaluation of the transformed convective terms has not been performed. These terms are treated explicitly in time and using a pseudo-spectral approach described in Section 4.5.

4.4.1 Nodal Matrix Equation

As with the 2D solver, a node-by-node ordering of the DOFs is used, in which all velocity and pressure unknowns at a node are ordered one after the other to form a small 8-component vector. This ordering results in a block-structured system matrix. We can now speak in terms of the small characteristic matrix equation associated with each node. The resulting nodal matrix equation is similar to its 2D counterpart, with two exceptions. First, there exist some new “Helmholtz coefficients,” A_1 and A_2 , that arise due to derivatives taken in the z -direction. Also, the nodal stiffness matrix is no longer 3×3 , but rather 8×8 . This is a consequence of the additional w -velocity component as well as that the Fourier coefficients have an imaginary and a real component that must both be computed.

To assist in implementation, the matrix equation is written in terms of the change in flow parameters during each time step,

$$\delta \vec{u}_k = \vec{u}_k^{n+1} - \vec{u}_k^n \quad (4.30)$$

$$\delta \hat{w}_k = \hat{w}_k^{n+1} - \hat{w}_k^n \quad (4.31)$$

$$\delta \hat{p}_k = \hat{p}_k^{n+1} - \hat{p}_k^n, \quad (4.32)$$

rather than the flow parameters themselves, \vec{u}_k , \hat{w}_k , and \hat{p}_k . Using \Re to denote the real component and \Im to denote the imaginary, the 3D nodal stiffness matrix, with unknowns ordered $\Re(\delta \hat{u}_k)_j$, $\Im(\delta \hat{u}_k)_j$, $\Re(\delta \hat{v}_k)_j$, $\Im(\delta \hat{v}_k)_j$, $\Re(\delta \hat{w}_k)_j$, $\Im(\delta \hat{w}_k)_j$, $\Re(\delta \hat{p}_k)_j$, $\Im(\delta \hat{p}_k)_j$, is

$$\begin{bmatrix} \mathcal{K}_{ij} & 0 & 0 & 0 & 0 & \frac{1}{2}^x q_{ij} & 0 \\ 0 & \mathcal{K}_{ij} & 0 & 0 & 0 & 0 & \frac{1}{2}^x q_{ij} \\ 0 & 0 & \mathcal{K}_{ij} & 0 & 0 & \frac{1}{2}^y q_{ij} & 0 \\ 0 & 0 & 0 & \mathcal{K}_{ij} & 0 & 0 & \frac{1}{2}^y q_{ij} \\ 0 & 0 & 0 & 0 & \mathcal{K}_{ij} & 0 & A_{2ij} \\ ^{xx} Q_{ij} & 0 & ^{xy} Q_{ij} & 0 & 0 & A_{2ij} & ^q S_{ij} \\ 0 & ^{yx} Q_{ij} & 0 & ^{yy} Q_{ij} & -A_{2ij} & 0 & 0 \\ 0 & ^{yy} Q_{ij} & -A_{2ij} & 0 & 0 & 0 & ^q S_{ij} \end{bmatrix} \begin{Bmatrix} \Re(\delta \hat{u}_k)_j \\ \Im(\delta \hat{u}_k)_j \\ \Re(\delta \hat{v}_k)_j \\ \Im(\delta \hat{v}_k)_j \\ \Re(\delta \hat{w}_k)_j \\ \Im(\delta \hat{w}_k)_j \\ \Re(\delta \hat{p}_k)_j \\ \Im(\delta \hat{p}_k)_j \end{Bmatrix} = - \left\{ \vec{\mathcal{D}}_k \right\}_i^n + \frac{3}{2} \left\{ \vec{\mathcal{C}}_k \right\}_i^n - \frac{1}{2} \left\{ \vec{\mathcal{C}}_k \right\}_i^{n-1} \quad (4.33)$$

where

$$\left\{ \vec{\hat{D}}_k \right\}_i^n = \begin{bmatrix} \mathcal{K}_{Dij} & 0 & 0 & 0 & 0 & 0 & {}^xq_{ij} & 0 \\ 0 & \mathcal{K}_{Dij} & 0 & 0 & 0 & 0 & {}^xq_{ij} & 0 \\ 0 & 0 & \mathcal{K}_{Dij} & 0 & 0 & 0 & {}^yq_{ij} & 0 \\ 0 & 0 & 0 & \mathcal{K}_{Dij} & 0 & 0 & 0 & {}^yq_{ij} \\ 0 & 0 & 0 & 0 & \mathcal{K}_{Dij} & 0 & 0 & A_{2ij} \\ 0 & 0 & 0 & 0 & 0 & \mathcal{K}_{Dij} - A_{2ij} & 0 & 0 \\ {}^{xx}\mathcal{Q}_{Dij} & 0 & {}^{xy}\mathcal{Q}_{Dij} & 0 & 0 & A_{2ij} & {}^qS_{ij} & 0 \\ 0 & {}^{yx}\mathcal{Q}_{Dij} & 0 & {}^{yy}\mathcal{Q}_{Dij} - A_{2ij} & 0 & 0 & 0 & {}^qS_{ij} \end{bmatrix} \begin{Bmatrix} \Re(\hat{u}_k)_j \\ \Im(\hat{u}_k)_j \\ \Re(\hat{v}_k)_j \\ \Im(\hat{v}_k)_j \\ \Re(\hat{w}_k)_j \\ \Im(\hat{w}_k)_j \\ \Re(\hat{p}_k)_j \\ \Im(\hat{p}_k)_j \end{Bmatrix}^n \quad (4.34)$$

and

$$\mathcal{K}_{ij} = \frac{1}{\Delta t} m_{ij} + \frac{1}{2} k_{ij} - \frac{1}{2} A_{1ij} \quad (4.35)$$

$${}^{xx}\mathcal{Q}_{ij} = -{}^xq_{ji} + \frac{1}{\Delta t} {}^{mx}S_{ij} + \frac{1}{2} {}^{kx}S_{ij} + \frac{1}{2} {}^{A_1x}S_{ij} \quad (4.36)$$

$${}^{xy}\mathcal{Q}_{ij} = -{}^xq_{ji} + \frac{1}{\Delta t} {}^{my}S_{ij} + \frac{1}{2} {}^{ky}S_{ij} + \frac{1}{2} {}^{A_1y}S_{ij} \quad (4.37)$$

$${}^{yx}\mathcal{Q}_{ij} = -{}^yq_{ji} + \frac{1}{\Delta t} {}^{mx}S_{ij} + \frac{1}{2} {}^{kx}S_{ij} + \frac{1}{2} {}^{A_1x}S_{ij} \quad (4.38)$$

$${}^{yy}\mathcal{Q}_{ij} = -{}^yq_{ji} + \frac{1}{\Delta t} {}^{my}S_{ij} + \frac{1}{2} {}^{ky}S_{ij} + \frac{1}{2} {}^{A_1y}S_{ij} \quad (4.39)$$

$$\mathcal{K}_{Dij} = k_{ij} - A_{1ij} \quad (4.40)$$

$${}^{xx}\mathcal{Q}_{Dij} = -{}^xq_{ji} + {}^{kx}S_{ij} + {}^{A_1x}S_{ij} \quad (4.41)$$

$${}^{xy}\mathcal{Q}_{Dij} = -{}^xq_{ji} + {}^{ky}S_{ij} + {}^{A_1y}S_{ij} \quad (4.42)$$

$${}^{yx}\mathcal{Q}_{Dij} = -{}^yq_{ji} + {}^{kx}S_{ij} + {}^{A_1x}S_{ij} \quad (4.43)$$

$${}^{yy}\mathcal{Q}_{Dij} = -{}^yq_{ji} + {}^{ky}S_{ij} + {}^{A_1y}S_{ij}. \quad (4.44)$$

The 'Helmholtz' and related coefficients are

$$A_{1ij} = -\nu(2\pi k/L)^2 m_{ij} \quad (4.45)$$

$${}^{A_1x}S_{ij} = -\nu(2\pi k/L)^2 {}^{mx}S_{ij} \quad (4.46)$$

$${}^{A_1y}S_{ij} = -\nu(2\pi k/L)^2 {}^{my}S_{ij} \quad (4.47)$$

$$A_{2ij} = (2\pi k/L) m_{ij}. \quad (4.48)$$

The other coefficients appearing in (4.33) are the same as for the 2D case. That is, the Galerkin terms are

$$m_{ij} = \int_{\Omega_e} N_i N_j d\Omega_e \quad (4.49)$$

$$k_{ij} = -\nu \int_{\Omega_e} \frac{\partial N_i}{\partial x} \frac{\partial N_j}{\partial x} + \frac{\partial N_i}{\partial y} \frac{\partial N_j}{\partial y} d\Omega_e \quad (4.50)$$

$${}^x q_{ij} = \int_{\Omega_e} \frac{\partial N_i}{\partial x} N_j d\Omega_e \quad (4.51)$$

$${}^y q_{ij} = \int_{\Omega_e} \frac{\partial N_i}{\partial y} N_j d\Omega_e \quad (4.52)$$

and the PSPG terms are

$${}^{mx} S_{ij} = \tau_{PSPG} \int_{\Omega_e} \frac{\partial N_i}{\partial x} N_j d\Omega_e \quad (4.53)$$

$${}^{my} S_{ij} = \tau_{PSPG} \int_{\Omega_e} \frac{\partial N_i}{\partial y} N_j d\Omega_e \quad (4.54)$$

$${}^{kx} S_{ij} = \int_{\Omega_e} \int_{\Omega} \tau_e \frac{\partial N_i}{\partial x} \nabla^2 N_j d\Omega_e \quad (4.55)$$

$${}^{ky} S_{ij} = \int_{\Omega_e} \tau_e \frac{\partial N_i}{\partial y} \nabla^2 N_j d\Omega_e \quad (4.56)$$

$${}^q S_{ij} = \tau_{PSPG} \int_{\Omega_e} \frac{\partial N_i}{\partial x} \frac{\partial N_j}{\partial x} + \frac{\partial N_i}{\partial y} \frac{\partial N_j}{\partial y} d\Omega_e. \quad (4.57)$$

4.4.2 Analytical Coefficient Evaluation

Because linear triangle elements are used, each of the coefficients appearing in the system matrix (4.33) can be evaluated analytically. The 'Helmholtz' and related coefficients become

$$A_{1ij} = -\nu(2\pi k/L)^2 \frac{S_T}{12} (1 + \delta_{ij}) \quad (4.58)$$

$$A_{1x} S_{ij} = -\tau_{PSPG} \nu(2\pi k/L)^2 \frac{{}^x n_i}{6} \quad (4.59)$$

$$A_{1y} S_{ij} = -\tau_{PSPG} \nu(2\pi k/L)^2 \frac{{}^y n_i}{6} \quad (4.60)$$

$$A_{2ij} = (2\pi k/L) \frac{S_T}{12} (1 + \delta_{ij}). \quad (4.61)$$

The Galerkin terms are

$$m_{ij} = \frac{S_T}{12} (1 + \delta_{ij}) \quad (4.62)$$

$$k_{ij} = \frac{\nu}{4S_T} ({}^x n_i {}^x n_j + {}^y n_i {}^y n_j) \quad (4.63)$$

$${}^x q_{ij} = -\frac{{}^x n_i}{6} \quad (4.64)$$

$${}^y q_{ij} = -\frac{{}^y n_i}{6}, \quad (4.65)$$

and the PSPG terms are

$${}^{mx} S_{ij} = \tau_{PSPG} \frac{{}^x n_i}{6} \quad (4.66)$$

$${}^{my}S_{ij} = \tau_{PSPG} \frac{{}^y n_i}{6} \quad (4.67)$$

$${}^{kx}S_{ij} = 0 \quad (4.68)$$

$${}^{ky}S_{ij} = 0 \quad (4.69)$$

$${}^q S_{ij} = \tau_{PSPG} \frac{1}{S_T} ({}^x n_i {}^x n_j + {}^y n_i {}^y n_j), \quad (4.70)$$

where S_T is the area of the triangle element, and δ is the Kronecker delta operator. The PSPG stabilization of the diffusion terms, ${}^{kx}S_{ij}$ and ${}^{ky}S_{ij}$, are zero due to the inability of the P1/P1 element to possess non-zero second derivatives.

4.4.3 Fourier Symmetry

The need to solve for both real and complex components of the Fourier coefficients adds much additional computational cost and detracts from the appeal of the spectral decomposition. However, this cost is compensated for in two ways. First, the higher accuracy of the spectral decomposition allows one to employ fewer nodes in the transverse direction than if a FE or FD scheme were used. Also, because it is known that the physical-space variables are purely real, there are simplifications that can be made to reduce the amount of computation required to solve the system. These simplifications come about because when the physical-space values, g_k , are purely real, the following symmetry appears in Fourier space:

$$\hat{g}(-f) = [\hat{g}(f)]^*, \quad (4.71)$$

where $[]^*$ denotes the complex conjugate. Consequently, the matrix equation need only be solved for the first $N/2 + 1$ Fourier modes, and the other modes can be evaluated from the symmetry condition. In addition, modes 0 and $N/2$ are purely real and the imaginary components can therefore be eliminated from the system. These simplifications are summarized in Figure 4.1.

4.5 Treatment of the Convective Terms

4.5.1 Temporal Treatment

As with the 2D solver, the convective terms are treated explicitly in the 3D solver. In addition to removing the nonlinearity from the matrix equation within each time step, it is seen from equations (4.12)–(4.14) or (4.19)–(4.22) that the Fourier modes are also completely decoupled within each time step since all coupling takes place in the convective terms. This is a monumental computational advantage because it converts the 3D problem into a series of independent 2D problems. The extension of the 2D code developed in Chapter 2 to 3D is rather direct because the basic building blocks (2D stabilized FE solvers) are already in place.

$k = 0$	$\rightarrow real$
$k = 1$	
\vdots	
$k = N/2 - 1$	$\} \rightarrow real + imaginary$
$k = N/2$	$\rightarrow real$
$k = N/2 + 1$	
\vdots	
$k = N - 1$	$\} \rightarrow \hat{g}_k = [\hat{g}_{N-k}]^*$

Fig. 4.1 – Illustration of the symmetry properties of the Fourier coefficients when the physical-space values are purely real.

4.5.2 Pseudo-Spectral Approach

Thus far the evaluation of the convective terms has not been discussed. The most straight-forward approach is to directly evaluate the discrete sums appearing in (4.23)–(4.29). However, this approach is computationally expensive because the discretization of at least $\mathcal{O}(N)$ products must be evaluated for each Fourier component, where N is the number of discrete Fourier modes. This results in $\mathcal{O}(N^2)$ product evaluations for each degree of freedom.

Alternatively, a pseudo-spectral approach first introduced by Orzag (Orzag, 1969, 1971a) is employed. In this approach, the weighted residual form of the convective terms are evaluated in physical space and then transformed back into Fourier space for use in the next time step. Referring to Figure 4.2, knowing the Fourier components of the velocity DOFs $(\hat{u}_k, \hat{w}_k)_i^{new}$, one can easily obtain the physical-space quantities $(\vec{u}_n, w_n)_i^{old}$ at N equi-spaced transverse points via a discrete inverse Fourier transform. Knowing the physical space velocity DOFs, the convective terms $(\vec{H}_n, H_{zn}, H_{pn})_i^{old}$ can then be computed using a combination of finite-elements and finite-differences. Upon evaluation of the convective terms at each 2D plane, their corresponding Fourier coefficients $(\hat{H}_k, \hat{H}_{zk}, \hat{H}_{pk})_i^{old}$ are computed using a DFT. Assuming there exist 2^n transverse planes, the terms can now evaluated at a cost of three fast Fourier transforms (FFTs) and $\mathcal{O}(n)$ products, resulting in $\mathcal{O}(N \log_2 N)$ product evaluations for each DOF.

The procedure for calculating the physical space convective terms $(\vec{H}_n, H_{zn}, H_{pn})_i$ knowing the velocity DOFs at N equi-spaced transverse planes is now explained in detail. First, the terms that must be evaluated are

$$(\vec{H}_n)_i = (H_{xn}, H_{yn})_i, \quad (4.72)$$

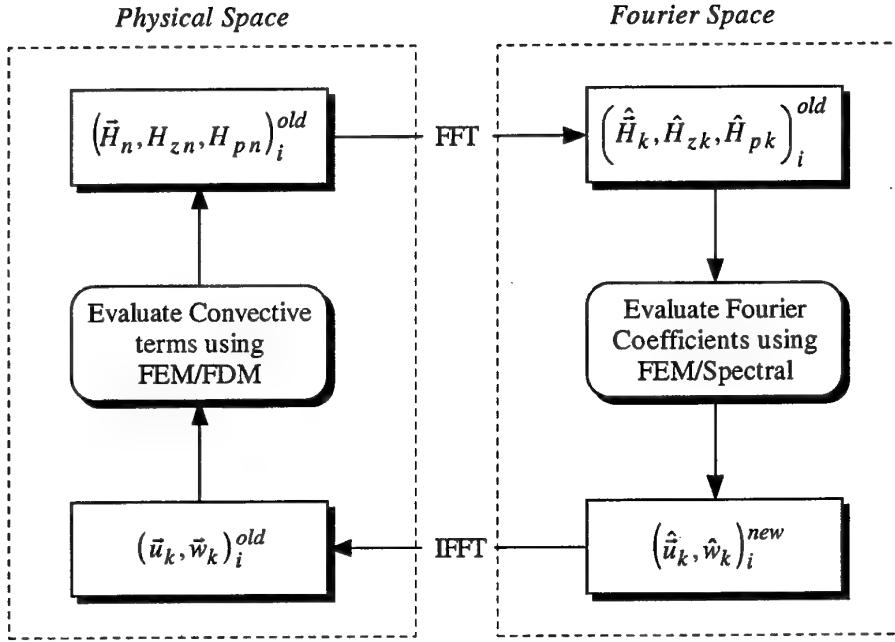


Fig. 4.2 – Illustration of the pseudo-spectral approach utilized in evaluation of the nonlinear terms.

where

$$(H_{xn})_i = (h_{xn})_i + (ST_{SUPG} x n)_i \quad (4.73)$$

$$(H_{yn})_i = (h_{yn})_i + (ST_{SUPG} y n)_i \quad (4.74)$$

and

$$(H_{zn})_i = (h_{zn})_i + (ST_{SUPG} z n)_i \quad (4.75)$$

$$(H_{pn})_i = (ST_{PSPG} C n)_i. \quad (4.76)$$

The components of the above expressions are

$$(h_{xn})_i = \int_{\Omega_e} N_i \left((\vec{u}_n \cdot \hat{\nabla}) u_n + w_n \left(\frac{\partial u}{\partial z} \right)_n \right) d\Omega_e \quad (4.77)$$

$$(h_{yn})_i = \int_{\Omega_e} N_i \left((\vec{u}_n \cdot \hat{\nabla}) v_n + w_n \left(\frac{\partial v}{\partial z} \right)_n \right) d\Omega_e \quad (4.78)$$

$$(h_{zn})_i = \int_{\Omega_e} N_i \left((\vec{u}_n \cdot \hat{\nabla}) w_n + w_n \left(\frac{\partial w}{\partial z} \right)_n \right) d\Omega_e \quad (4.79)$$

$$(ST_{SUPGxn})_i = \tau_{SUPG} \int_{\Omega_e} \vec{u}_n \cdot \tilde{\nabla} N_i \left(\frac{\partial u_n}{\partial t} + (\vec{u}_n \cdot \tilde{\nabla}) u_n + w_n \left(\frac{\partial u}{\partial z} \right)_n + \frac{\partial p_n}{\partial x} - \nu \left(\frac{\partial^2 u}{\partial z^2} \right)_n \right) d\Omega_e \quad (4.80)$$

$$(ST_{SUPGyn})_i = \tau_{SUPG} \int_{\Omega_e} \vec{u}_n \cdot \tilde{\nabla} N_i \left(\frac{\partial v_n}{\partial t} + (\vec{u}_n \cdot \tilde{\nabla}) v_n + w_n \left(\frac{\partial v}{\partial z} \right)_n + \frac{\partial p_n}{\partial y} - \nu \left(\frac{\partial^2 v}{\partial z^2} \right)_n \right) d\Omega_e \quad (4.81)$$

$$(ST_{SUPGzn})_i = \tau_{SUPG} \int_{\Omega_e} \vec{u}_n \cdot \tilde{\nabla} N_i \left(\frac{\partial w_n}{\partial t} + (\vec{u}_n \cdot \tilde{\nabla}) w_n + w_n \left(\frac{\partial w}{\partial z} \right)_n + \left(\frac{\partial p}{\partial z} \right)_n - \nu \left(\frac{\partial^2 w}{\partial z^2} \right)_n \right) d\Omega_e \quad (4.82)$$

$$(ST_{PSPGcn})_i = \tau_{PSPG} \int_{\Omega_e} \tilde{\nabla} N_i \cdot \left((\vec{u}_n \cdot \tilde{\nabla}) \vec{u}_n + w_n \left(\frac{\partial \vec{u}}{\partial z} \right)_n \right) d\Omega_e. \quad (4.83)$$

The transverse derivatives $(\partial/\partial z)_n$, $(\partial^2/\partial z^2)_n$ are calculated using second-order central differences. That is,

$$\left(\frac{\partial f}{\partial z} \right)_n = \frac{f_{right} - f_{left}}{2\Delta z} \quad (4.84)$$

$$\left(\frac{\partial^2 f}{\partial z^2} \right)_n = \frac{f_{right} - 2f_n + f_{left}}{\Delta z^2}, \quad (4.85)$$

where Δz is the spacing between the 2D planes,

$$\Delta z = \frac{L}{N} \quad (4.86)$$

and L is the transverse dimension of the computational domain and N is the number of Fourier modes. The values of f_{left} and f_{right} are

$$f_{left} = f_{n-1}, \quad f_{right} = f_{n+1} \quad (4.87)$$

at all but the boundary planes $n = 0$ and $n = N - 1$. At these planes the periodicity condition results in

$$f_{left} = f_{N-1}, \quad f_{right} = f_1 \quad (4.88)$$

for the $n = 0$ plane and

$$f_{left} = f_{N-2}, \quad f_{right} = f_0 \quad (4.89)$$

for the $n = N - 1$ plane. These spanwise derivative values calculated from (4.84) and (4.85) are then treated as any other nodal DOF in computing the weighted residual form of the convective terms.

4.5.3 Nodal Convective Load Vector

Referring to (4.33), the characteristic nodal matrix equation associated with each node, the nodal convective load vector is

$$\left\{ \vec{\mathcal{C}}_k \right\}_i = \sum_{n=0}^{N-1} \left\{ \vec{C}_n \right\}_i e^{-\frac{2\pi I_n}{N} k}, \quad (4.90)$$

where

$$\left\{ \vec{C}_n \right\}_i = \begin{Bmatrix} (H_{xn})_i \\ (H_{yn})_i \\ (H_{zn})_i \\ (H_{pn})_i \end{Bmatrix} \quad (4.91)$$

and $(H_{xn})_i$, $(H_{yn})_i$, $(H_{zn})_i$, and $(H_{pn})_i$ are given in (4.73)–(4.76).

4.5.4 Analytical Coefficient Evaluation

Because linear triangle elements are used, each of the coefficients appearing in the convective load vector (4.91) can be easily evaluated analytically. After integration of (4.77)–(4.83) one obtains

$$(h_{xn})_i = \frac{u_{nj}}{24} (1 + \delta_{ik}) (u_{nk}^x n_j + v_{nk}^y n_j) + \frac{S_T}{60} \max(1, 2[\delta_{ij} + \delta_{ik} + \delta_{jk}]) w_{nk} \left(\frac{\partial u}{\partial z} \right)_{nj} \quad (4.92)$$

$$(h_{yn})_i = \frac{v_{nj}}{24} (1 + \delta_{ik}) (u_{nk}^x n_j + v_{nk}^y n_j) + \frac{S_T}{60} \max[1, 2(\delta_{ij} + \delta_{ik} + \delta_{jk})] w_{nk} \left(\frac{\partial v}{\partial z} \right)_{nj} \quad (4.93)$$

$$(h_{zn})_i = \frac{w_{nj}}{24} (1 + \delta_{ik}) (u_{nk}^x n_j + v_{nk}^y n_j) + \frac{S_T}{60} \max[1, 2(\delta_{ij} + \delta_{ik} + \delta_{jk})] w_{nk} \left(\frac{\partial w}{\partial z} \right)_{nj} \quad (4.94)$$

$$(ST_{SUPG xn})_i = \tau_{SUPG} (u_{nk}^x n_i + v_{nk}^y n_i) \left[\frac{1}{24} (1 + \delta_{jk}) \frac{\delta u_j}{\Delta T} + \frac{1}{12 S_T} u_{nj} (u_{nk}^x n_j + v_{nk}^y n_j) + \frac{1}{24} (1 + \delta_{jk}) \left(w \frac{\partial u}{\partial z} \right)_{nj} + \frac{x_{nj}}{12 S_T} p_{nj} - \frac{\nu}{24} (1 + \delta_{jk}) \left(\frac{\partial^2 u}{\partial z^2} \right)_{nj} \right] \quad (4.95)$$

$$(ST_{SUPG \text{ } yn})_i = \tau_{SUPG} (u_{nk}^x n_i + v_{nk}^y n_i) \left[\frac{1}{24} (1 + \delta_{jk}) \frac{\delta v_j}{\Delta T} + \frac{1}{12S_T} v_{nj} (u_{nk}^x n_j + v_{nk}^y n_j) + \frac{1}{24} (1 + \delta_{jk}) \left(w \frac{\partial v}{\partial z} \right)_{nj} + \frac{v_{nj}}{12S_T} p_{nj} - \frac{\nu}{24} (1 + \delta_{jk}) \left(\frac{\partial^2 v}{\partial z^2} \right)_{nj} \right] \quad (4.96)$$

$$(ST_{SUPG \text{ } zn})_i = \tau_{SUPG} (u_{nk}^x n_i + v_{nk}^y n_i) \left[\frac{1}{24} (1 + \delta_{jk}) \frac{\delta w_j}{\Delta T} + \frac{1}{12S_T} w_{nj} (u_{nk}^x n_j + v_{nk}^y n_j) + \frac{1}{24} (1 + \delta_{jk}) \left(w \frac{\partial w}{\partial z} \right)_{nj} + \frac{1}{24} (1 + \delta_{jk}) \left(\frac{\partial p}{\partial z} \right)_{nj} - \frac{\nu}{24} (1 + \delta_{jk}) \left(\frac{\partial^2 w}{\partial z^2} \right)_{nj} \right] \quad (4.97)$$

$$(ST_{PSPG \text{ } Cn})_i = \tau_{PSPG} \frac{1}{24} (u_{nk}^x n_j + v_k^y n_j) (u_{nj}^x n_i + v_j^y n_i) + \frac{1}{24} (1 + \delta_{jk}) \left({}^x n_i \left(\frac{\partial u}{\partial z} \right)_{nj} + {}^y n_i \left(\frac{\partial v}{\partial z} \right)_{nj} \right), \quad (4.98)$$

where S_T is the area of the triangle element and δ is the Kronecker delta operator.

4.6 Parallelization

Though LES is a compromise between RANS and DNS, it is still rather computationally expensive due to the fine mesh and small time steps required. In the case of the solver developed in this work, where an unstructured mesh is utilized, parallelization is required in order to reduce the computation time to a reasonable level.

In this section, the parallelization theory and implementation are discussed. There are three sections of the algorithm that are parallelized. Referring to Figure 4.10, these sections are 1) evaluation of Fourier coefficients, 2) evaluation of convective terms, and 3) transformation of flowfield information to/from Fourier space. The parallelization theory as it applies to each of these sections will now be presented. Following which, the implementation in terms of OpenMP on a shared-memory computer is discussed.

4.6.1 Parallelization Theory

4.6.1.1 Evaluation of Fourier Coefficients

The majority of the computation time is spent setting up and solving the linear systems that arise for each of the Fourier modes at each time step. Consequently, parallelization of this portion of the algorithm is most important and beneficial.

The typical approach taken to parallelize solvers of this type is to partition the compu-

tational mesh in physical space, as shown in Figure 4.3(a). That is, the mesh is divided into NP subregions (where NP is the number of processors available), and the work associated with each subregion (a submatrix in the overall system matrix) assigned to different threads. One drawback to this approach is that the partitioning procedure can be complicated and expensive. Another is that the convergence characteristics and accuracy can change depending upon the partitioning. Yet another is that the submatrices are coupled to one another through the boundaries of the subregions, and information must therefore be passed between processors during the solution of the system.

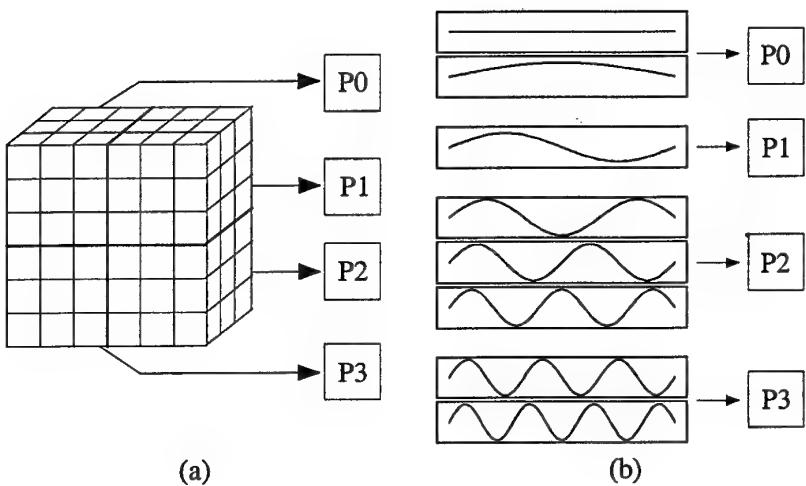


Fig. 4.3 – Approaches to partitioning for parallelization: (a) partitioning in physical space, (b) partitioning in Fourier space.

Because the discretization procedure used in this work produces an independent matrix equation for each Fourier mode rather than a single large coupled system, a novel approach is taken. Rather than partitioning in physical space, the problem is partitioned in Fourier space. As illustrated in Figure 4.3(b), this means that each Fourier mode is treated as a different thread and can be calculated on a different processor. It is important to understand that rather than using a parallel matrix equation solver as is typically done, NP instances of a serial matrix solver, each solving a different Fourier mode, are called simultaneously.

This approach is free from the drawbacks of the physical space partitioning mentioned above. First, the partitioning process is no longer complicated or expensive. In fact, it is virtually free because it occurs as a natural by-product of the FE/spectral discretization technique. In addition, because the partitions (Fourier modes) are completely decoupled within each time step, accuracy and convergence are not affected by parallelization. Also due to the decoupling of the partitions, there is no need for communication between processors during the system solution.

The parallelization technique is not without its drawbacks, however. As mentioned previously, though the linear system solution takes place in Fourier space, the convective terms are treated with a pseudo-spectral approach, meaning these terms are evaluated in physical space. Consequently, the flowfield must first be passed through an inverse DFT. This requires that at the end of each time step, all flowfield information from all Fourier modes (all processors) is required in order to calculate the convective terms needed for the following time step. Passing this amount of data between processors on a distributed-memory machine is cost-prohibitive. Another drawback is that load imbalancing can be an issue. Load imbalancing occurs because a parallel region will not complete until all processors finish their work. If some processors have more work to complete than others, performance suffers. In this algorithm, load imbalancing is a problem because there are relatively few “work chunks” (Fourier modes) with possibly varying computational costs. That is, there are only $N/2 + 1$ linear systems to solve (where N is the number of Fourier modes) and, depending upon the level to which a particular mode is excited, a widely varying amount of time may be required to solve each system. Both of these drawbacks must be addressed in order to successfully implement this parallelization scheme.

4.6.1.2 Evaluation of the Convective Terms

The parallelization concept for this section of the solver is similar to that of the Fourier coefficient evaluation, except that the computations are no longer in Fourier space, but in physical space. The problem is partitioned by considering each 2D plane as a separate task. These tasks are coupled in that values at the two neighboring planes are required for the FD evaluation of the transverse derivatives. However, this coupling is of an explicit nature because the evaluation of the convective terms at one plane is only dependent on previously known values at the neighboring planes, not the evaluated convective terms at those planes. Consequently, any plane can be solved by any thread and in any order.

4.6.1.3 Forward/Inverse Fourier Transform

Each node in the 2D plane results in an independent FFT/IFFT procedure. Though parallel FFT algorithms are common, their use is inappropriate for this application. Instead, for this section of the solver the work is partitioned in the 2D plane, similar to Figure 4.3(a). In this way, NP instances of a serial FFT/IFFT routine are called simultaneously, and the transforms are preformed in a node-by-node manner within each partition. This procedure does not suffer from any of the accuracy issues mentioned previously for physical space partitioning because each node can be transformed independently. Also, there are no complicated partitioning algorithms required. In fact, the assignment of a particular node to a particular thread can be completely random...the partitions need not be contiguous blocks.

4.6.2 Parallelization Implementation

The parallelization is implemented using the OpenMP model (Still et al., 1998; Adve et al., 1999; Throop, 1999) because, as will be shown in this section, its features allow the drawbacks discussed above (especially in the Fourier coefficient evaluation section) to be addressed. OpenMP is a collection of directives, runtime library routines, and environment variables for shared-memory parallelism using Fortran77/90/95 or C/C++. It is designed to be a vendor-independent standard, and its development is overseen by a board consisting of members representing most of the major computer manufacturers. Industry support for this standard has been shown from companies such as Compaq, Hewlett-Packard, Intel, IBM, SiliconGraphics, and Sun. Consequently, it is quickly becoming the standard for shared-memory parallelism.

The difference between a shared-memory computer and a distributed-memory system is important to understand when developing a parallelization strategy. The primary distinction is that for a shared-memory computer, all of the processors are able to directly access all of the memory in the machine. Though there are a number of architectures designed for shared-memory parallelism, from the OpenMP programmer's point of view, these details are unimportant and the system can be understood sufficiently from the simplified architecture shown in Figure 4.4. The exception to this is in trying to tune a code for optimal performance on a particular computer; in this case the detailed architecture is important.

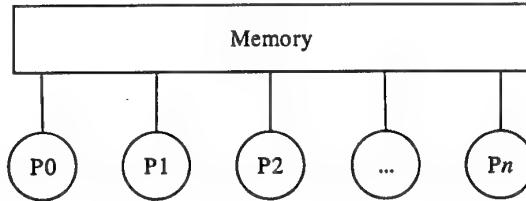


Fig. 4.4 – A simplified model of the shared-memory computer architecture, where each of the n processors has direct access to all system memory.

Distributed-memory systems are the alternative to shared-memory configurations. In these systems, each processor is only capable of directly addressing memory that is physically associated with it, as seen from the simplified architecture shown in Figure 4.5. Because each processor has its own independent memory system, the programmer must manage how the program data is distributed to them. In accessing data stored in memory connected to another processor, the programmer must explicitly pass messages through the interconnecting network. This is typically done using message-passing libraries such as Message Passing Interface (MPI) (Message Passing Interface Forum, 1995; Pacheco, 1996) and Parallel Virtual Machine (PVM) (Geist et al., 1994).

Both types of parallel machines have advantages over the other. In terms of the shared-

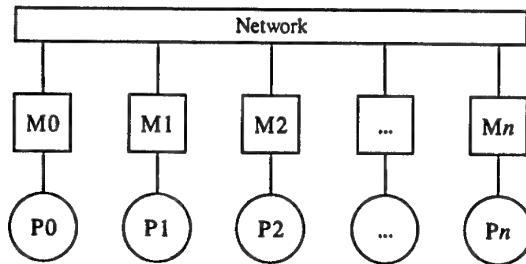


Fig. 4.5 – A simplified model of the distributed-memory computer architecture, where each of the n processors has direct access only to its own memory and access to memory located on another processor requires passing information through the interconnecting network.

memory computer, these advantages are

- Incremental Parallelization. One has the ability to easily parallelize small parts of an application at a time, while leaving the remaining parts unmodified. This is important when one or more small parts of the code take large fractions of the computational time. In this case, a significant application speedup can be obtained at a small cost in terms of coding time.
- Ease of Implementation. Shared-memory parallelization is generally easier to code because less of the details need be addressed by the programmer.
- Small Coding Overhead. Very small increases in code volume are required over the serial version. Typical values are 2-25% code volume increase (Chandra et al., 2001). Furthermore, in the case of OpenMP, parallel codes will compile and run on serial machines with no code changes.

The drawbacks to shared-memory machines are

- Scalability. Typically, shared-memory machines are not scalable to massive numbers of processor in the way that distributed-memory systems are.
- Specialized Compiler. A special compiler and set of run-time libraries are required.
- Expense. Though shared-memory computers have become much more affordable, especially in the last decade, they are often more expensive than a similar (in terms of processors and memory) distributed-memory system.

From the above discussion it is apparent that the algorithm developed in this work requires a shared-memory computer. This is because, as mentioned in the previous section,

all flowfield information from all threads is required in order to calculate the convective terms needed to advance a time step. Using a distributed memory system, all of this data would have to pass through the interconnecting network, which is not feasible considering the speed of current network technology. On a shared-memory computer, however, all the threads can access the same memory locations and the parallelization approach becomes tractable.

An additional advantage to shared-memory machines is that the scheduling of work chunks to processors need not be fixed. Take for example a problem in which one has six work chunks to be solved on three processors. For static load scheduling, each work chunk is fixed to a specific processor, as shown in Figure 4.6. This type of scheduling is required with distributed-memory systems. Using a shared-memory machine, one has the option of implementing dynamic load scheduling. This approach allows for better load balancing because threads that finish early “ask” for more work. In terms of the example problem, this concept is illustrated in Figure 4.7. At the beginning of the parallel region, Figure 4.7(a), all work chunks are pooled together. Each thread is then given one (or more, if desired) work chunk to solve, Figure 4.7(b). In this example, work chunk W0 requires much more time to solve than W1 and W2. Consequently, processors P1 and P2 finish their work first and are given another work chunk, Figure 4.7(c). Processor P2 is the first processor to become free, and is given the last work chunk to complete, Figure 4.7(d). This dynamic load scheduling is applied to the parallelization of the Fourier coefficient evaluation section of the code. The other parallel regions are well balanced and hence a static scheduling is used.

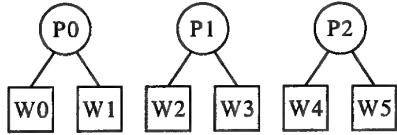


Fig. 4.6 – Static load scheduling: assignment of work chunks to processors is fixed and determined prior to execution.

OpenMP uses a fork-join model for parallelization. As shown in Figure 4.8, the program initially executes as a single thread (called the master thread), which spawns a team of threads when a parallel region is encountered. At the end of a parallel region, all additional threads are terminated and the master thread continues execution. It is because of this approach that parallelism can be added incrementally. Two basic flavors of parallel regions are available: segmentation of the code into different sections to be run as different threads, and execution of the iterations of a do-loop in parallel.

Initiation and specification of parallel regions, as well as other OpenMP-related tasks, is accomplished by introducing directives into the serial source code. In free-form Fortran90/95 source code, a line that begins with the sentinel

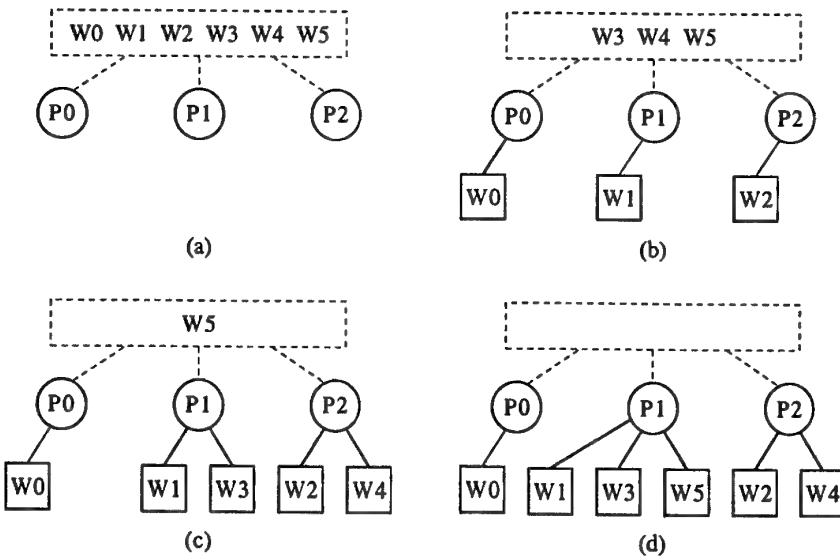


Fig. 4.7 – Dynamic load scheduling: assignment of work chunks to processors occurs dynamically during execution on a first-completed-first-served basis.

```
!$omp ...
```

is treated as an OpenMP directive by an OpenMP compiler. Because the directives appear as comment lines to non-OpenMP compilers, correctly written code can be compiled and run on a serial machine with no code changes.

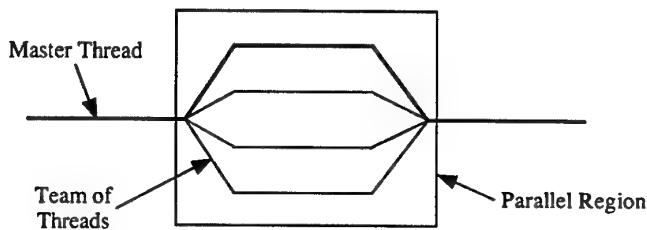


Fig. 4.8 – Fork-join parallelism model: the master thread spawns a team of threads inside a parallel region.

The parallel regions in the solver developed in this work are all of the do-loop flavor. In order to successfully parallelize a do-loop, the loop must have no dependencies. That is, the result of any operation in a loop iteration cannot depend on the result of any other iteration. This means that the iterations can be executed in any order and/or by any processor. The Fortran90 syntax for parallelizing a do-loop is

```

!$omp parallel do
do ...
...
...
end do
!$omp end parallel do

```

The opening directive must appear immediately before the start of the `do`-loop, and the closing directive must appear immediately after the associated `end do` statement. Additional directives are typically included in order to specify those variables that are private to each thread and those that are shared among all threads, to control how the work is divided among the processors, to initialize private variables, etc. For example, consider the following subroutine to perform a single Jacobi iteration in solving the Laplace equation on a rectangular domain with Dirichlet boundary conditions using finite-differences. It is assumed that the boundary condition values are already been assigned in the `a` array and that the spacing between nodes is constant and equal in both directions. The serial routine is

```

subroutine jacobi_it(a,a_old,nx,ny)
integer :: nx,ny,i,j
real, dimension(nx,ny) :: a,a_old
do i=2,nx-1
  do j=2,ny-1
    a(i,j)=0.25*(a_old(i+1,j)+a_old(i-1,j)+a_old(i,j+1)+a_old(i,j-1))
  enddo
enddo
return
end

```

Parallelization of this routine using the `parallel do` directive results in

```

subroutine jacobi_it(a,a_old,nx,ny)
integer :: nx,ny,i,j
real, dimension(nx,ny) :: a,a_old
!$omp parallel do private(i,j) shared(a,a_old,nx,ny)
do i=2,nx-1
  do j=2,ny-1
    a(i,j)=0.25*(a_old(i+1,j)+a_old(i-1,j)+a_old(i,j+1)+a_old(i,j-1))
  enddo
enddo

```

```

!$omp end parallel do
return
end

```

The **private** clause specifies those variables that should be kept private to each thread. In this case, each thread keeps its own copy of the counter variables *i* and *j*. The **shared** clause, on the other hand, specifies those variables that are shared among all threads.

The explanation of OpenMP given above is, of course, simplified and incomplete. However, it is sufficient for understanding of the parallel implementation of the algorithm developed in this work. For further information on programming with OpenMP, the reader is referred to the book by Chandra et al. (Chandra et al., 2001) or the official OpenMP specifications (ope, 2000, 1998).

4.6.3 Parallelization Performance

The performance of the parallelization procedure can be ascertained via a test case. The algorithm described in this chapter is used to solve flow past a circular cylinder at $Re = 195$. The details of this particular problem will be discussed in detail in the following chapter. The in-plane mesh consists of approximately 17,000 nodes, and 32 Fourier modes are used in transverse direction. The code is run on an SGI Origin2000 computer with 8 195 MHz R10000 processors and approximately 2.5Gb of RAM. To ascertain the parallel performance, the simulation is run for 100 time steps using 1, 2, ..., 8 processors. Figure 4.9 shows the solver speedup (the computation time divided by the serial computation time) as a function of the number of processors used. The major reason the actual speedup (SFE-3D) falls away from the ideal is the previously-mentioned problem of load imbalancing. Though the dynamic load scheduling helps remedy this, the work chunks are so large and varying that it is common for the solver to be forced to wait while the last work chunk is completed. It should be noted that as more Fourier modes are used, the parallelization becomes more efficient. Of course, it is now apparent that the parallelization is not scalable to massive numbers of processors. In fact, because there are only $N/2 + 1$ work chunks, typically using more than $N/4$ processors is not cost effective and the use of more than $N/2 + 1$ processors is virtually useless.

4.7 SFE3D Summary

This section provides a detailed summary of the implementation of the FE/spectral algorithm described previously in this chapter. The resulting solver is hereby referred to as SFE3D.

SFE3D seeks to find an approximate solution to the 3D unsteady NS equations in terms of the primitive variables (u, v, w, p) . When the governing equations (4.3)–(4.5) are discretized using the FE/spectral method, a system of linear equations is formed for each Fourier mode.

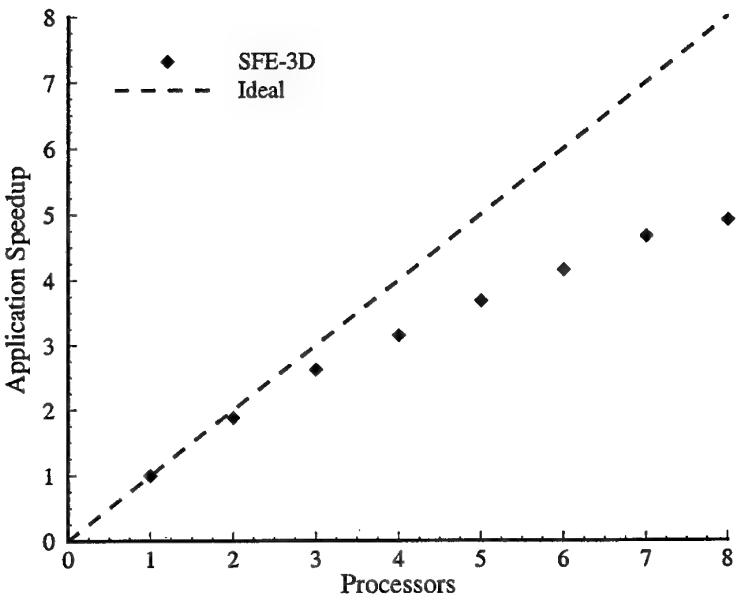


Fig. 4.9 – Application speedup as a function of the number of processors used.

When solved, the result is an approximation for the Fourier coefficients of the primitive variables $(\hat{u}, \hat{v}, \hat{w}, \hat{p})$. Because SFE3D uses a nodal ordering of the unknowns, when this linear system is written in matrix form, the matrix equations to be solved can be characterized by an 8×8 system associated with each node at each Fourier mode (4.33).

All the unknowns appearing in (4.33) at each Fourier mode are obtained by looping through the nodes of each 2D element and summing the contributions. Each of the coefficient contributions are obtained from analytical expressions derived from the combined FE/spectral method. These expressions are given in (4.58)–(4.70) and (4.92)–(4.98).

Figure 4.10 presents a flow chart giving the basic structure of the SFE3D solver. Each of the blocks in this chart will now be discussed in some detail.

4.7.0.0.1 Boundary and initial conditions.

Boundary condition information read from the input grid file is processed, and Dirichlet values are stored at all appropriate nodes. The initial conditions—the values of \vec{u}, w, p for time step levels n and $n - 1$ —are set to the values stored in the input grid file if a restart is requested. Otherwise, the initial conditions are set to zero. The convective terms at time steps n and $n - 1$ are also computed at this time (see the paragraph “Evaluation of convective terms” for more details).

4.7.0.0.2 Create and store $[A]_k$ matrices.

The system matrix for each Fourier mode $[A]_k$ is computed via the FE/spectral method described in this chapter. These matrices are constant in time and hence need only be computed once at the beginning of a run and then stored for later use. These matrices are stored in the sparse matrix format described in Appendix A in the single shared array $a(:, k)$.

4.7.0.0.3 Create and store preconditioning matrices.

Because the system matrices are constant, so too are the preconditioning matrices. Hence, these matrices need only be computed once at the beginning of a run and then stored for later use. In addition, because the system matrices associated with each Fourier mode are very similar, preconditioning matrices need not be computed for every mode. Instead, preconditioning matrices computed for one mode can be applied to the neighboring modes. This approach saves memory and does so at very little cost in terms of computational speed. These preconditioning matrices are stored in the sparse matrix format described in Appendix A in the shared arrays $pa(:, k)$, $pia(:, k)$, and $pja(:, k)$.

4.7.0.0.4 Create system matrix structure.

The structure of the sparse system matrices (the location of the non-zero entries) is determined from the mesh connectivity information read from the modified DPlot input file. The structures of the system matrices for each Fourier mode are identical. Consequently, the structure is only computed once and stored in the shared arrays $ia(:)$ and $ja(:)$ described in Appendix A.

4.7.0.0.5 Evaluation of convective terms.

In this parallelized block, the convective terms are calculated for each node on each of the N equi-spaced 2D planes. A parallel do-loop is performed through the 2D planes. For each plane, the following steps are completed. 1) The $\partial/\partial z$ and $\partial^2/\partial z^2$ values are calculated for the DOFs at each node using finite-differences and are stored in temporary arrays. 2) The convective terms (including the stabilization terms described in Section 4.5) are calculated using finite-elements, treating the $\partial/\partial z$ and $\partial^2/\partial z^2$ terms as known DOFs at each node.

4.7.0.0.6 FFT (boundary and initial conditions).

The boundary and initial conditions, which are given in physical space, are needed in Fourier space for use in the evaluation of the Fourier coefficients. This transformation is accomplished via a node-by-node FFT in the transverse direction.

4.7.0.0.7 FFT (convective terms).

The convective terms, which are calculated in physical space, must be transformed into Fourier space for use in the evaluation of the new Fourier coefficients. This transformation is accomplished via a node-by-node FFT in the transverse direction.

4.7.0.0.8 FFT (flowfield values).

The flowfield values at the new time step must be transformed into physical space for output and/or evaluation of the convective terms. This transformation is accomplished via a node-by-node IFFT in the transverse direction.

4.7.0.0.9 Fourier coefficient evaluation.

It is inside this parallelized block that most of the computation time is spent. The decoupling of the Fourier modes allows a parallel do-loop to be executed through the Fourier modes. For each mode, the following steps are completed. 1) The RHS vector $\mathbf{b}(:,k)$ is filled as controlled by (4.33) and the boundary conditions. 2) The matrix equation is solved using the SPARSKIT package (see Section 2.4). The zero vector is used as the initial guess. This is the natural choice because the solution $\{\phi\}$ is the change in flow variables between time steps and these values are expected to be small. 3) The solution for each mode is stored in the shared array $\text{phi}(:,k)$.

4.7.0.0.10 Initialization.

The user-defined parameters for SFE3D are given via a Fortran NAMELIST file. These parameters contain such information as time step size, domain transverse dimension, number of time steps, number of threads, sparse matrix solver parameters, and Dirichlet boundary condition values. Additionally, input and output filenames are obtained and the corresponding files are opened.

4.7.0.0.11 Read grid/restart file.

The input grid file is read. The format of this file is a modified form of the DPlot format, which was developed at the von Karman Institute and is described in Appendix C. This file contains the number of 2D planes (Fourier modes), the 2D nodal coordinates and connectivity, boundary condition types, and flowfield values at each node and plane for time steps n and $n - 1$. After this file has been read, one-time calculations such as element areas and normals are calculated. Also, memory is allocated for all arrays used in the solver.

4.7.0.0.12 Update time step.

Once all of the matrix equations have been solved for the current time step, the flow variables are updated for the next time step. That is,

$$\begin{aligned} \left(\vec{\hat{u}}_i\right)_k^{n-1} &= \left(\vec{\hat{u}}_i\right)_k^n, & (\hat{w}_i)_k^{n-1} &= (\hat{w}_i)_k^n, & (\hat{p}_i)_k^{n-1} &= (\hat{p}_i)_k^n \\ \left(\vec{\hat{u}}_i\right)_k^n &= \left(\vec{\hat{u}}_i\right)_k^{n+1}, & (\hat{w}_i)_k^n &= (\hat{w}_i)_k^{n+1}, & (\hat{p}_i)_k^n &= (\hat{p}_i)_k^{n+1} \\ \left(\vec{\hat{u}}_i\right)_k^{n+1} &= \left(\vec{\hat{u}}_i\right)_k^{n+1} + \delta \left(\vec{\hat{u}}_i\right)_k, & (\hat{w}_i)_k^{n+1} &= (\hat{w}_i)_k^{n+1} + \delta (\hat{w}_i)_k, & (\hat{p}_i)_k^{n+1} &= (\hat{p}_i)_k^{n+1} + \delta (\hat{p}_i)_k \end{aligned}$$

4.7.0.0.13 Write data files.

The Fortran NAMELIST file contains an output frequency parameter stating the number of time steps between writing of the output files. If the current time step requests an output, a restart file is written in the modified DPlot format defined in Appendix C as well as a post-processing file in TecPlot and/or FieldView unstructured grid format.

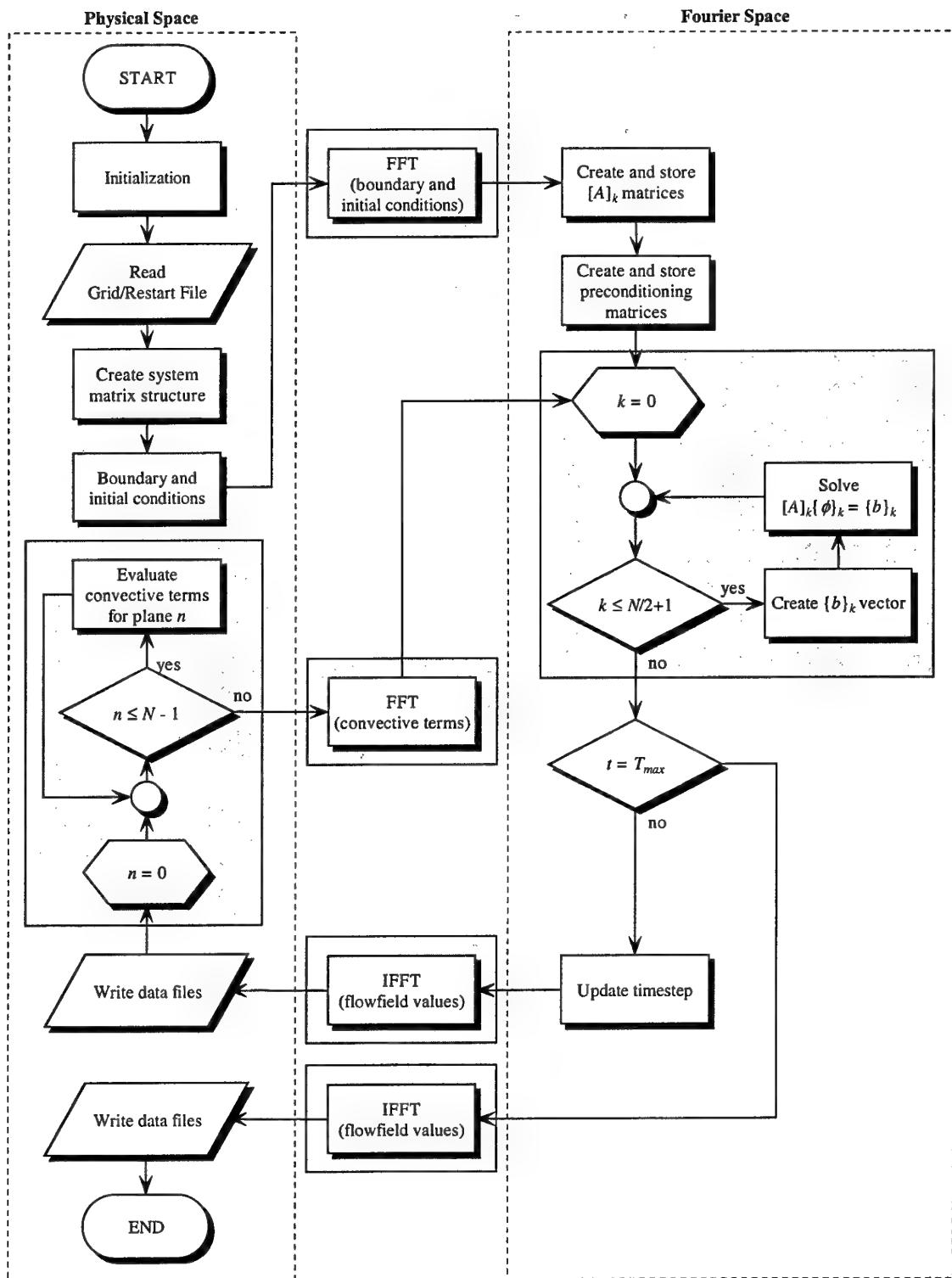


Fig. 4.10 – Flow chart showing the basic structure of the SFE3D Navier-Stokes solver. Sections of the code that are parallelized are denoted by a shaded box.

5. 3D LAMINAR FLOW TEST CASE

This chapter presents a validation test case for the SFE3D solver. Specifically, the case of flow past a circular cylinder in the 3D transition regime is chosen because the flow is laminar and yet there exist 3D features with distinct characteristics that vary with Reynolds number.

5.1 Circular Cylinder Problem Description

A short discussion on circular cylinder flow has already been presented in Section 3.3. As mentioned in that section, the flow over a circular cylinder is interesting because it contains many distinct flow characteristics, including boundary layers, separation, shear layers, and a wake region. For this reason, and also because a vast amount of experimental and numerical data exists for comparison, circular cylinder flow is an excellent test case for validating NS solvers.

The behavior of the flow is highly dependent upon the Reynolds number,

$$Re_D = \frac{U_\infty D}{\nu}, \quad (5.1)$$

where U_∞ is the freestream velocity, D the cylinder diameter, and ν the kinematic viscosity. For $Re \lesssim 180$ the wake behind the cylinder is 2D, while at $Re \gtrsim 180$ the wake is unstable to transverse perturbations and develops 3D structures. Simulation results for the 2D regime were presented in Section 3.3. In this chapter, Reynolds numbers of 195 and 300 are presented. These Reynolds numbers are both in the 3D regime, but well below the onset of turbulence, which occurs at $Re \approx 1200$.

According to Williamson (Williamson, 1996b), at $Re \approx 180 - 200$, the wake behind the cylinder develops streamwise structures of rather large dimension. These structures are known as mode-A instabilities and typically have wavelengths on the order of $3-4D$. The $Re = 195$ case chosen for this study lies in this regime. As the Reynolds number is increased to around 260, 3D structures with wavelengths on the order of $1D$, known as mode-B instabilities, become dominant. The $Re = 300$ case chosen for this study is in this regime. Figure 5.1, taken from the experiments of Williamson (Williamson, 1996a), illustrates the mode-A and mode-B instabilities via dye streaks in the wake.

The qualitative correctness of the SFE3D solver can be verified by its ability to produce mode-A and mode-B instabilities at the correct Reynolds numbers. In addition, some quantitative checks are performed for the $Re = 300$ case. Specifically, the shedding frequency in terms of the Strouhal number,

$$St = \frac{fD}{U_\infty}, \quad (5.2)$$

is compared with experiments. Also, the mean velocity profiles in the wake are compared with experimental and numerical results. The mean values of interest are the streamwise

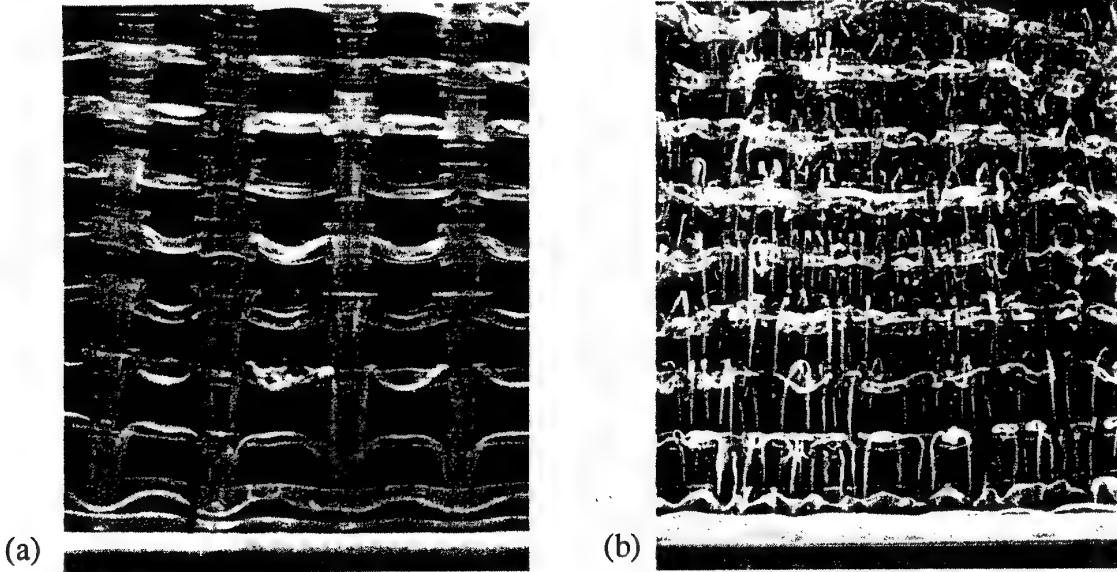


Fig. 5.1 – 3D circular cylinder flow: mode-A and mode-B instabilities in the cylinder wake, as visualized by Williamson (Williamson, 1996b). (a) shows mode-A instabilities at $Re = 200$ with spanwise wavelength $\lambda/D = 4.01$. (b) shows mode-B instabilities at $Re = 300$ with spanwise wavelength $\lambda/D \approx 1$.

velocity \bar{u} , crossflow velocity \bar{v} , streamwise velocity fluctuations

$$\overline{u'^2} = \overline{uu} - \bar{u}^2, \quad (5.3)$$

crossflow velocity fluctuations

$$\overline{v'^2} = \overline{vv} - \bar{v}^2, \quad (5.4)$$

and the Reynolds shear stress

$$\overline{u'v'} = \overline{uv} - \bar{u}\bar{v}. \quad (5.5)$$

5.2 Computational Mesh

The computational domain used for both the $Re = 195$ and $Re = 300$ cases is displayed in Figure 5.2. A Dirichlet velocity inlet condition is applied $5D$ upstream of the cylinder center, while a Neumann outlet condition exists $20D$ downstream. As reported by Kravchenko and Moin (Kravchenko and Moin, 1998), the shedding frequency does not become independent of the lateral domain size until approximately $60D$. In order to keep the computations tractable, however, the lateral domain extends only $7D$ from the cylinder in this study. At these extents a slip wall condition is applied. Due to the proximity of the walls, a slightly higher shedding frequency is expected compared to free-stream results. Mittal and Balachandar (Mittal and

Balachandar, 1995) studied the sensitivity of numerical simulations to transverse domain size at $Re = 300$. They showed that if the domain is small, variations in the mean flow parameters (lift, drag, shedding frequency) appear. Considering this study, as well as that of Kravchenko and Moin (Kravchenko and Moin, 1998), a spanwise dimension of $2\pi D$ was deemed sufficient. Of course, as mentioned in Chapter 4, periodic boundary conditions exist on these spanwise boundaries.

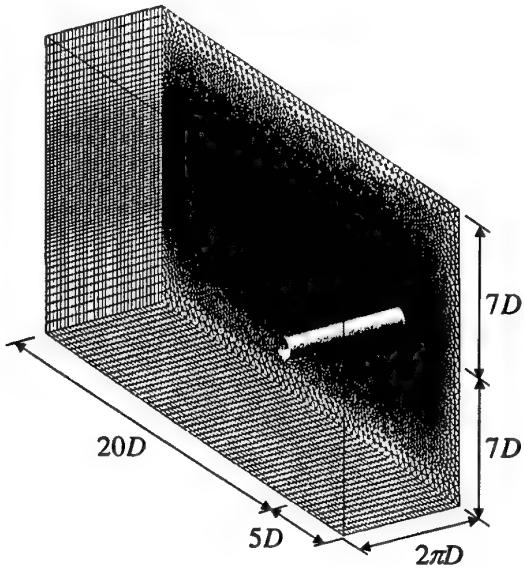


Fig. 5.2 – 3D circular cylinder flow: overview of the computational domain and mesh containing 33,000 in-plane triangle elements (17,000 in-plane nodes) and 32 Fourier modes.

In the 2D plane, the mesh consists of 33,000 triangle elements with approximately 17,000 nodes. As seen in Figure 5.2, nodes are clustered in the wake region, with characteristic element side lengths of approximately $0.15D$. Outside the wake region the mesh is more coarse, with the characteristic element side length being $0.5D$ at the upstream and lateral domain extents. Elements are also clustered toward the cylinder in order to better capture the initial development of the shed vorticity. Boundary layer wedge-type elements are used very near the cylinder walls, with the perpendicular dimension of the first element off the cylinder being $0.005D$. The smallest spanwise structures are expected at $Re = 300$ and have a wavelength on the order of $1D$. Because a spectral method is used in this direction, 16 modes are sufficient to capture these scales. However, in order to verify that smaller scales are not dominant, 32 modes are used... allowing wavelengths as small as $0.4D$ to be captured. Overall, the mesh contains approximately 544,000 nodes.

The mesh described above is certainly more refined than necessary. Consequently, to check for grid convergence a coarse mesh simulation was also performed for the $Re = 300$ case.

In this mesh, the resolution is essentially halved in each direction, resulting approximately 7,400 in-plane elements (3,800 in-plane nodes) and 16 Fourier modes.

5.3 Results

For each case ($Re = 195$, $Re = 300$, $Re = 300$ coarse mesh), the simulation was initialized using potential flow. To decrease the computation time, the solution was allowed to develop as a 2D flow until the Karman vortex street was fully developed. In addition, to hasten the asymmetric shedding, a time-dependent rotation was applied to the cylinder at the beginning of the simulation for a short time; first in one direction, then in the opposite. After the 2D Karman vortex street had developed, a small perturbation was applied at the inlet boundary for a short time to initiate the 3D features of the flow.

At $Re = 195$ and $Re = 300$, the cylinder wake is unstable to the spanwise perturbation, and the perturbation should grow to develop the mode-A and mode-B streamwise vortex structures. At lower Reynolds numbers, however, the wake is stable to spanwise perturbations. As a first check on the SFE3D solver, a $Re = 100$ case was run. As expected, the perturbation simply diminished as it was propagated downstream and eventually washed out of the domain. Figure 5.3 shows instantaneous surfaces of constant transverse vorticity $tU_\infty/D = 40$ after termination of the inlet perturbation. The flow no longer has any 3D features—only the zeroth Fourier mode is non-zero.

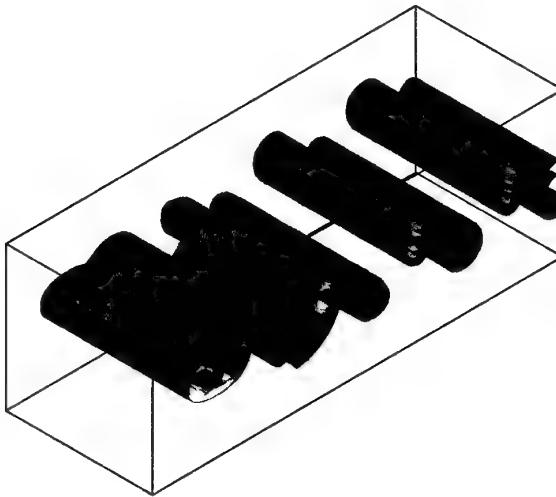


Fig. 5.3 – 3D circular cylinder flow: surfaces of constant transverse vorticity showing the 2D Karman vortex street at $Re = 100$ after the transverse perturbation imposed at the inlet has washed downstream.

For the $Re = 195$ case, the perturbation does indeed persist and grow in the wake

region. Eventually, a statistically steady 3D Karman vortex street develops. Figure 5.4 shows the qualitatively correct mode-A instabilities. The yellow and cyan surfaces mark constant positive and negative streamwise vorticity, and highlight the alternate vortex shedding. The blue and red surfaces mark constant positive and negative transverse vorticity and show the mode-A structures with characteristic wavelength $\sim 3D$. These results are in excellent agreement with the computations of Thompson et al (Thompson et al., 1994) shown in Figure 5.5.



Fig. 5.4 – 3D circular cylinder flow: mode-A instabilities in the wake at $Re = 195$. The red and blue surfaces mark a positive and negative value of streamwise vorticity, and the yellow and cyan surfaces mark a positive and negative value of transverse vorticity.

Figure 5.6 shows the qualitatively correct mode-B instabilities in the cylinder wake as predicted by SFE3D. The yellow and cyan surfaces again mark a constant positive and negative streamwise vorticity, and highlight the alternate vortex shedding. The blue and red surfaces mark constant positive and negative values of spanwise vorticity and show the mode-B structures with characteristic wavelength $\sim 1D$. Again, these results are in excellent qualitative agreement with the computations of Thompson et al (Thompson et al., 1994), shown in Figure 5.7.

As a quantitative check on the results, the flow statistics are plotted and compared with the published numerical results of Kravchenko and Moin (Kravchenko and Moin, 1998), who used a B-spline method with zonal grids. These results, in turn, agree well with the spectral results of Mittal and Balachandar (Mittal and Balachandar, 1997). Figures 5.8–5.12 show velocity statistics at five streamwise locations in the wake. The SFE3D statistics are averaged in the transverse direction and over approximately eight shedding cycles. From these figures it is seen that the fine-grid SFE3D results are in good agreement with the B-spline results for both the mean velocities and the time-averaged fluctuations. The slight discrepancy between the SFE3D and B-spline results has two likely causes. The first is that this author



Fig. 5.5 – 3D circular cylinder flow: mode-A instabilities in the wake, as calculated by Thompson et al (Thompson et al., 1994). The light and dark surfaces mark a particular value of positive and negative streamwise vorticity, and the other surface marks a value of spanwise vorticity.

was unable to acquire tabular data from either Kravchenko or Moin. Instead, the B-spline profiles were digitized from published plots, which introduces error. The second probable cause is the proximity of the lateral wall boundaries. In the B-spline simulations, a far-field condition was imposed $30D$ from the cylinder, whereas the SFE3D simulations utilize a slip-wall condition located $7D$ from the cylinder. The nearness of the walls causes an additional acceleration of the flow as it passes the cylinder, resulting in increased recirculation and higher velocity fluctuations. Coarse-grid SFE3D results are also shown in Figures 5.8–5.12. These results are sufficiently close to the fine-grid, and convergence to the published values is achieved with grid refinement.

The time-averaged drag coefficient for the cylinder at $Re = 300$ is computed using (3.4), averaged over the span. The shedding frequency is determined by computing the power spectrum of the time-accurate lift coefficient, and the Strouhal number subsequently evaluated from equation (5.2). The results are shown in Table 5.1 along with some experimental values and the numerical results of Kravchenko and Moin (Kravchenko and Moin, 1998). The drag coefficient predicted by SFE3D is slightly higher than that measured by Weiselsberger (Weiselsberger, 1922), but is sufficiently accurate and changes very little between the coarse and fine meshes. The Strouhal number did not change with grid refinement, and is slightly higher than the value measured by Williamson (Williamson, 1996b). This is to be expected, however, due to the proximity of the lateral walls in the SFE3D simulations.

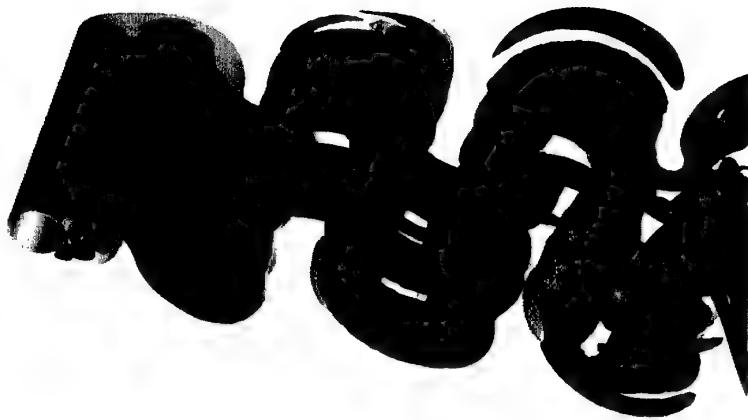


Fig. 5.6 – 3D circular cylinder flow: mode-B instabilities in the wake at $Re = 300$. The red and blue surfaces mark a positive and negative value of streamwise vorticity, and the yellow and cyan surfaces mark a positive and negative value of transverse vorticity.

Table 5.1 – 3D circular cylinder flow: comparison of C_D and St for SFE3D, published experiments (Weiselsberger, 1922; Williamson, 1996b), and published numerical results (Kravchenko and Moin, 1998) at $Re = 300$.

	C_D	St
SFE3D (coarse)	1.24	0.208
SFE3D (fine)	1.29	0.208
Williamson exp.	–	0.203
Weiselsberger exp.	1.22	–
B-Spline	1.28	0.203

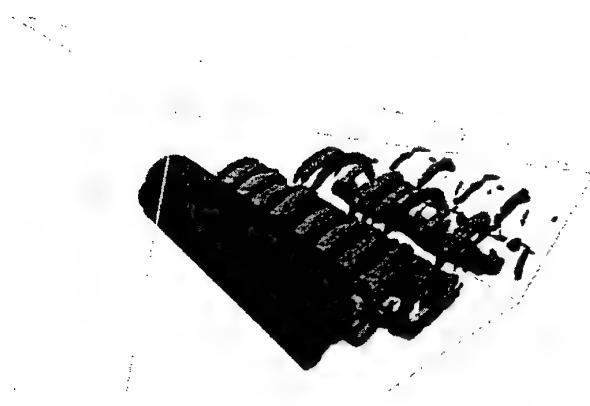


Fig. 5.7 – 3D circular cylinder flow: mode-B instabilities in the wake, as calculated by Thompson et al (Thompson et al., 1994). The light and dark surfaces mark a particular value of positive and negative streamwise vorticity, and the other surface marks a value of spanwise vorticity.

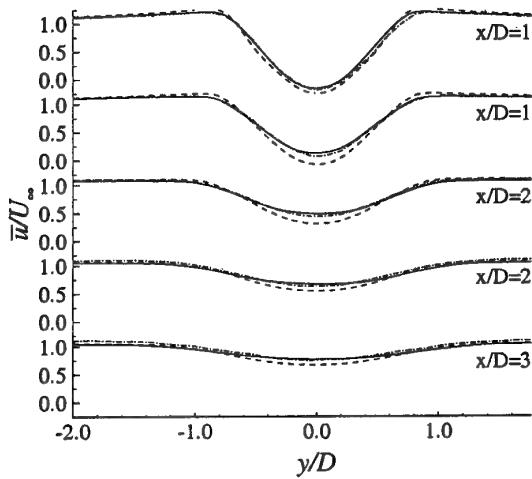


Fig. 5.8 – 3D circular cylinder flow: mean streamwise velocity at different streamwise locations in the wake at $Re = 300$. — SFELES fine grid, - - - SFELES coarse grid, - · - - B-spline simulations by Kravchenko and Moin (Kravchenko and Moin, 1998).

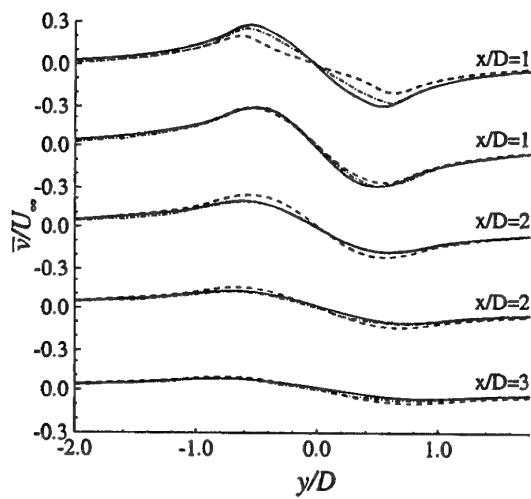


Fig. 5.9 – 3D circular cylinder flow: mean lateral velocity at different streamwise locations in the wake at $Re = 300$. — SFELES fine grid, - - - SFELES coarse grid, - · - B-spline simulations by Kravchenko and Moin (Kravchenko and Moin, 1998).

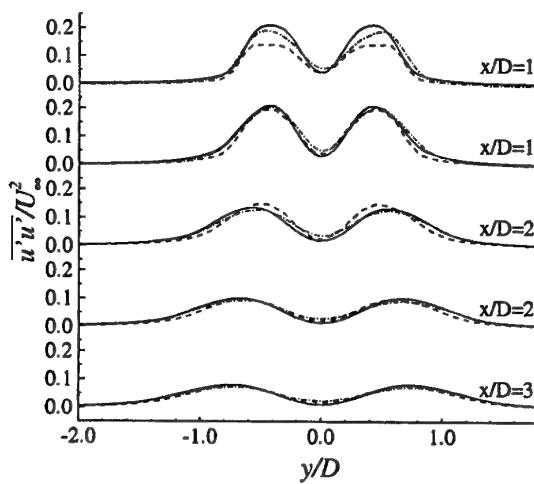


Fig. 5.10 – 3D circular cylinder flow: time-averaged streamwise velocity fluctuations at different streamwise locations in the wake at $Re = 300$. — SFELES fine grid, - - - SFELES coarse grid, - · - B-spline simulations by Kravchenko and Moin (Kravchenko and Moin, 1998).

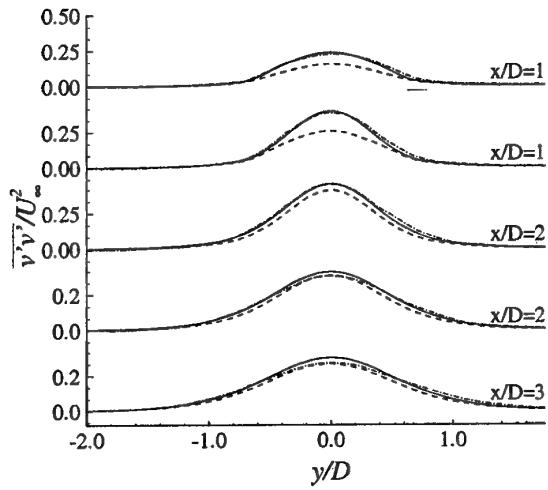


Fig. 5.11 – 3D circular cylinder flow: time-averaged lateral velocity fluctuations at different streamwise locations in the wake at $Re = 300$. — SFELES fine grid, - - - SFELES coarse grid, - · - - B-spline simulations by Kravchenko and Moin (Kravchenko and Moin, 1998).

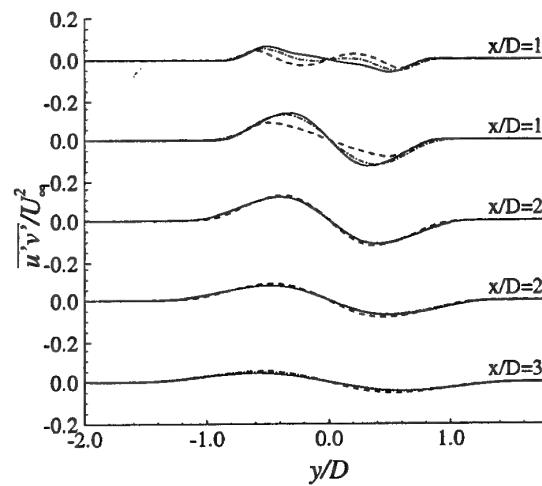


Fig. 5.12 – 3D circular cylinder flow: time-averaged Reynolds shear stress at different streamwise locations in the wake at $Re = 300$. — SFELES fine grid, - - - SFELES coarse grid, - · - - B-spline simulations by Kravchenko and Moin (Kravchenko and Moin, 1998).

6. LARGE-EDDY SIMULATION SOLVER

It is commonly felt that the pacing factor for the development of more accurate design and analysis tools for aerodynamic and industrial applications is the development of better models of turbulent flow. Since the first LES results were published by Deardorff in 1970 (Deardorff, 1970), great advances have been made both in LES algorithms and available computer hardware. In its infancy, LES was limited to simple flows and to the very few groups that could afford access to supercomputers. It was developed as a compliment to laboratory experiments investigating the fundamental physics of simple turbulent flows. With the advances in computer technology, the possibility of using LES is now available to many, and can be applied to more complex flows. However, LES is still limited mainly to rather simple geometries that can be meshed via Cartesian or other structured meshes.

In this chapter the extension of the 3D solver developed in Chapter 4 to LES is discussed. The LES approach to handling turbulence is presented and compared with the RANS and DNS approaches. The governing equations are then derived and a review of subgrid scale models is given. Finally, the implementation procedure in terms of the FE/spectral method of this work is provided.

6.1 LES Approach to Turbulence

Simulations of turbulent flows are difficult because of the large range of scales involved: the largest scales are of the flow domain, while the smallest are on the order of the Kolmogorov length scale (Kolmogorov, 1941, 1962). The DNS approach to turbulence involves directly solving the NS equations with no modeling of the turbulence—all turbulent fluctuations are calculated explicitly as shown in Figure 6.1(a). Of course, because turbulence is an unsteady and 3D phenomenon, no 2D or steady-state simplifications can be made in solving the NS equations. As a consequence of needing to capture the small scale (high frequency) fluctuations, extremely fine meshes and time steps are required. In one dimension, the number of nodes required is $N_n \sim L/\eta$, where L is the dimension of the computational domain and η is the Kolmogorov length scale. This ratio scales as $Re^{3/4}$, consequently the number of nodes needed for DNS (which is 3D) scales as $N_n^3 \sim Re^{9/4}$. Taking into account the number of calculations per node and the number of time steps required to advance the solution for a sufficient time, the overall cost of a DNS computation is $\mathcal{O}(Re^3)$. In other words, doubling the Reynolds number means the computational effort is increased by at least a factor of eight! In addition, highly accurate (high-order) schemes must be used in order to reduce dispersion and dissipation errors. These schemes (spectral methods, for example) do not lend themselves well to complex geometries and/or boundary conditions. Due to current algorithm and (especially) hardware limitations, DNS is limited to simple geometries and very low Reynolds numbers; typically below those of engineering interest.

Simulations of high Reynolds number turbulent flows have been commonplace for the past three decades, however, as a result of utilizing the RANS equations. In this approach, one follows the original idea of Reynolds (Reynolds, 1895) in assuming the fluid is in a randomly unsteady turbulent state and, via time-averaging, all quantities can be considered as having a mean and a fluctuating part. The time average of a function is defined as

$$F = \frac{1}{T} \int_{t-T}^t f(\tau) d\tau, \quad (6.1)$$

where T is a time interval much longer than the largest fluctuation time scale. The function $f(t)$ can now be decomposed into its mean F and fluctuating $f(t) - F$ parts. Applying this averaging to the NS equations results in the RANS equations, which are solved only for the mean flow quantities while accounting for the effects of the fluctuations via a turbulence model. In this way, all unsteadiness is averaged out as being part of the turbulence, shown in Figure 6.1(b). RANS solutions tend to suffer from accuracy and generality problems, both of which are attributed to the shortcomings of the turbulence models. Models are typically ‘tuned’ by applying them to simple flows for which theoretical or thoroughly validated experimental results exist. When the models are applied to flows that differ from those by which they are tuned, adjusting of the model is required in order to obtain acceptable results. These shortcomings are a consequence of the model being required to represent such a wide range of scales. As stated by Piomelli (Piomelli, 1994), “While the small scales tend to depend only on viscosity, and may, therefore, be somewhat universal, the large ones are affected very strongly by the boundary conditions... Thus, it does not seem possible to model the effect of the large scales of turbulence in the same way in flows that are very different.”

The LES approach lies between the extremes of DNS, in which all fluctuations are resolved, and the RANS approach, in which only mean quantities are calculated and the fluctuations modeled. The basic philosophy, as illustrated in Figure 6.1(c), is to explicitly compute only the large-scale motions that are directly affected by the boundary conditions, and to model the effects of the small-scale motions. The driving notion behind this approach is that the small-scale motions are more isotropic and dependent primarily on viscosity effects; modeling of these terms should therefore be more general. In addition, the majority of the computational cost of DNS is spent in resolving the small-scale, dissipative motions, while the quantities of engineering interest can be calculated mainly from the large-scale motions. Because the large scales of turbulence are explicitly calculated, LES is inherently unsteady, 3D, and requires fairly fine meshes and time steps. However, it can be used at much higher Reynolds numbers than DNS. In fact, assuming the small scales ideally obey inertial range dynamics, and in the absence of solid boundaries, the cost of a computation is independent of Reynolds number.

The separation of the large scales from the small is accomplished via a low-pass filter operation. A filtered quantity, denoted in this work by an overbar, is defined in the manner

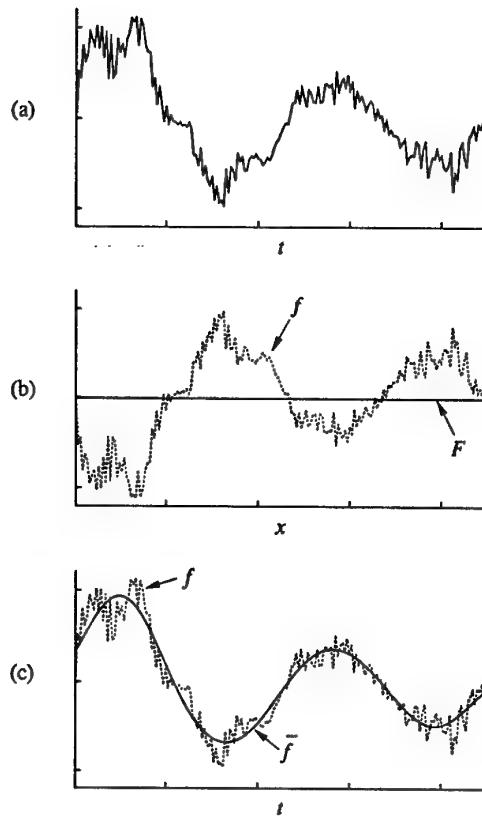


Fig. 6.1 – Comparison of the different approaches to handling turbulence. a) Direct Numerical Simulation (DNS), b) Reynolds Averaged Navier-Stokes (RANS), c) Large Eddy Simulation (LES).

introduced by Leonard (Leonard, 1974):

$$\bar{f}(\vec{x}, t) = \int_{\Omega} f(\vec{r}, t) G(\vec{x} - \vec{r}) d\vec{r}, \quad (6.2)$$

where Ω is the computational domain and G is the filter function. The filter function must satisfy the normalization condition

$$\int_{\Omega} G(\vec{x} - \vec{r}) d\vec{r} = 1. \quad (6.3)$$

It also should provide a mean-preserving filtering operation (for a constant filter width) that commutes with differentiation.

The most commonly used filter functions are the tophat filter,

$$G(x) = \begin{cases} 1/\Delta, & \text{if } |x| \leq \Delta/2 \\ 0, & \text{otherwise,} \end{cases} \quad (6.4)$$

shown in Figure 6.2a; the Gaussian filter,

$$G(x) = \sqrt{\frac{6}{\pi\Delta^2}} e^{-\frac{6x^2}{\Delta^2}}, \quad (6.5)$$

shown in Figure 6.2b; and the Fourier cutoff filter,

$$G(x) = \frac{\sin(\pi x/\Delta)}{\pi x}, \quad (6.6)$$

shown in Figure 6.2c. In the above expressions, Δ is the filter width.

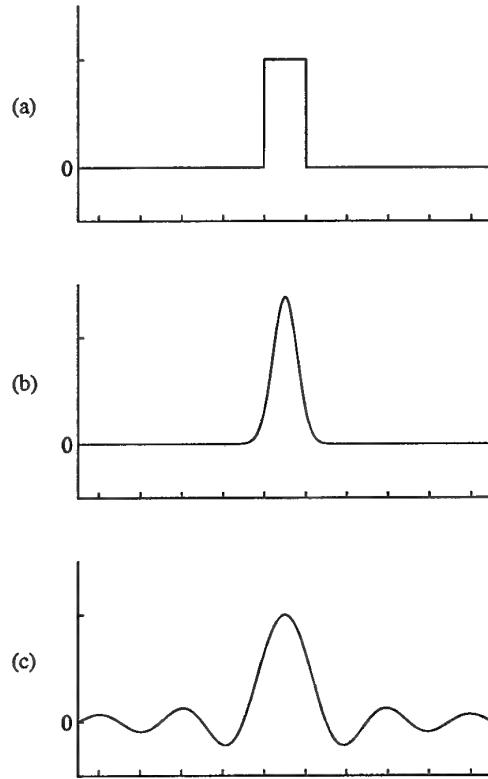


Fig. 6.2 – Commonly-used filter functions. a) tophat, b) Gaussian, and c) sharp Fourier cutoff.

Figure 6.3 shows how each of these filter functions performs when applied to a general 1D function. The dotted line shows the original function, the solid line is the filtered function, and the diamond symbols show the spacing of the sampling points. Frames (a)-(c) all utilize the same filter width, while frame (d) shows the effect of reducing the filter width by a factor of eight using the sharp Fourier cutoff filter function. It is seen that each filter function successfully filters out the high-frequency oscillations. It is also seen that the sharp Fourier

cutoff tends to better reproduce the smaller resolved scales. However, this added performance comes at the cost of being non-local in physical space.

Additional insight into these filter functions can be obtained by analyzing their properties in Fourier space. Given that $f(x)$ has a Fourier transform

$$\hat{f}(k) = \mathcal{F}[f(x)], \quad (6.7)$$

the Fourier transform of the filtered function is

$$\hat{\tilde{f}}(k) = \hat{G}(k)\hat{f}(k), \quad (6.8)$$

where $\hat{G}(k)$ is termed the transfer function and is 2π times the Fourier transform of the filter function

$$\hat{G}(k) = 2\pi\mathcal{F}[G(x)]. \quad (6.9)$$

The tophat transfer function is

$$\hat{G}(k) = \frac{\sin\left(\frac{1}{2}k\Delta\right)}{\frac{1}{2}k\Delta}, \quad (6.10)$$

the Gaussian transfer function is

$$\hat{G}(k) = e^{-\frac{k^2\Delta^2}{24}}, \quad (6.11)$$

and the sharp Fourier cutoff transfer function is

$$\hat{G}(k) = \begin{cases} 1, & \text{if } |k| \leq k_c \\ 0, & \text{otherwise,} \end{cases} \quad (6.12)$$

where k_c is the cutoff frequency

$$k_c = \frac{\pi}{\Delta}. \quad (6.13)$$

The characteristics of the filter functions in Fourier space can now be viewed in terms of their attenuation factors, $\hat{G}(k)^2$, shown in Figure 6.4. It is seen that the tophat filter is not effective in attenuating high wave numbers. In addition, the tophat and Gaussian filters not only attenuate the high frequencies (small scales), but also some of the low frequencies (large scales) as well.

6.2 Governing Equations

The LES governing equations are derived by applying the filtering operation (6.2) to the NS equations. The incompressible NS equations in the absence of body forces and written in tensor notation are

$$\frac{\partial u_i}{\partial t} + \frac{\partial}{\partial x_j}(u_i u_j) = -\frac{\partial p}{\partial x_i} + \nu \nabla^2 u_i \quad (6.14)$$

$$\frac{\partial u_i}{\partial x_i} = 0, \quad (6.15)$$

where u_i is the i th component of the velocity vector and p is the kinematic pressure (pressure divided by density). The particular filter function G used in arriving at the governing equations is immaterial, as long as it follows the rules given in the previous section. The filter operation (6.2) commutes with both temporal and spatial derivatives (see Appendix D). That is,

$$\frac{\partial \bar{f}}{\partial t} = \overline{\frac{\partial f}{\partial t}}, \quad (6.16)$$

and

$$\frac{\partial \bar{f}}{\partial x_i} = \overline{\frac{\partial f}{\partial x_i}}. \quad (6.17)$$

$$(6.18)$$

Using (6.17), filtering of the continuity equation (6.15) is straightforward:

$$\begin{aligned} \overline{\frac{\partial u_i}{\partial x_i}} &= 0 \\ \downarrow \\ \overline{\frac{\partial \bar{u}_i}{\partial x_i}} &= 0. \end{aligned} \quad (6.19)$$

The momentum equation (6.14) is equally straightforward using both (6.16) and (6.17):

$$\begin{aligned} \overline{\frac{\partial u_i}{\partial t}} + \overline{\frac{\partial}{\partial x_j}(u_i u_j)} &= -\overline{\frac{\partial p}{\partial x_i}} + \nu \nabla^2 \bar{u}_i \\ \downarrow \\ \overline{\frac{\partial \bar{u}_i}{\partial t}} + \overline{\frac{\partial}{\partial x_j}(\bar{u}_i \bar{u}_j)} &= -\overline{\frac{\partial \bar{p}}{\partial x_i}} + \nu \nabla^2 \bar{u}_i. \end{aligned} \quad (6.20)$$

The $\overline{u_i u_j}$ term poses a problem because it is the filtering operation applied to a product, and $\overline{u_i u_j} \neq \bar{u}_i \bar{u}_j$. The above expression is typically rewritten as

$$\overline{\frac{\partial \bar{u}_i}{\partial t}} + \overline{\frac{\partial}{\partial x_j}(\bar{u}_i \bar{u}_j)} = -\overline{\frac{\partial \bar{p}}{\partial x_i}} + \overline{\frac{\partial \tau_{ij}}{\partial x_j}} + \nu \nabla^2 \bar{u}_i, \quad (6.21)$$

where

$$\tau_{ij} = \bar{u}_i \bar{u}_j - \overline{u_i u_j}. \quad (6.22)$$

This term, which accounts for the effects of the unresolved scales, is called the subgrid-scale (SGS) Reynolds stress, and must be modeled via a SGS model.

With the exception of the SGS Reynolds stress term, the LES governing equations are identical in form to the laminar NS equations. Consequently (assuming that a suitable model for the SGS Reynolds stress exists) nearly identical solution algorithms can be employed.

6.3 Subgrid-Scale Models

The filtered momentum equations are closed via a SGS model that attempts to account for the effects of the small scales on the resolved scales. The primary task of the model is to correctly account for the energy transfer between the large and small scales. While on the average energy is transferred from the large scales to the subgrid scales, energy flow from the small to the large scales can occur intermittently. This phenomenon is called backscatter and is currently an area of interest for SGS model development. Most models in use, however, are absolutely dissipative—only allowing energy to flow into the small scales.

Though it is an area of intense research, this work does not attempt to further the technology of SGS models. Instead, the focus is on the development and implementation of the FE/spectral discretization algorithm. Consequently, the common Smagorinski model is implemented. In this section then, the majority of effort is spent discussing the Smagorinski model. Other common models are presented, but only briefly. The interested reader is referred to the references cited in conjunction with each model.

6.3.1 Smagorinski Model

The Smagorinski model, originally proposed in 1963 (Smagorinski, 1963), is certainly the most commonly-used SGS model. A Boussinesq approximation is first made, in which the SGS Reynolds stress is assumed proportional to the resolved strain:

$$\tau_{ij} = 2\nu_t \bar{S}_{ij} + \frac{1}{3}\tau_{\ell\ell}\delta_{ij}, \quad (6.23)$$

where

$$\bar{S}_{ij} = \frac{1}{2} \left(\frac{\partial \bar{u}_i}{\partial x_j} + \frac{\partial \bar{u}_j}{\partial x_i} \right) \quad (6.24)$$

is the strain rate tensor of the filtered field, and ν_t is termed the eddy viscosity. Models of this type are commonly referred to as eddy viscosity models.

Substituting the Boussinesq approximation (6.23) into the filtered momentum equation (6.21) gives

$$\frac{\partial \bar{u}_i}{\partial t} + \frac{\partial}{\partial x_j} (\bar{u}_i \bar{u}_j) = -\frac{\partial \bar{P}}{\partial x_i} + 2 \frac{\partial}{\partial x_j} [(\nu + \nu_t) \bar{S}_{ij}], \quad (6.25)$$

where

$$\bar{P} = \bar{p} - \frac{1}{3}\tau_{\ell\ell}, \quad (6.26)$$

and, as before, \bar{p} is the filtered kinematic pressure. From the incompressibility condition, (6.25) can be rewritten as

$$\frac{\partial \bar{u}_i}{\partial t} + \frac{\partial}{\partial x_j} (\bar{u}_i \bar{u}_j) = -\frac{\partial \bar{P}}{\partial x_i} + (\nu + \nu_t) \nabla^2 \bar{u}_i + 2 \frac{\partial \nu_t}{\partial x_j} \bar{S}_{ij}. \quad (6.27)$$

It is seen that, with the exception of the last term on the right-hand side, (6.27) is identical in form to the laminar momentum equation (4.1).

From dimensional analysis, ν_t should be proportional to a length scale times a velocity scale:

$$\nu_t \sim \ell \cdot q_{sgs}. \quad (6.28)$$

The most active resolved scales are those very near the filter width, consequently the natural length scale for LES is the filter width Δ . Smagorinski's model, based on a mixing length hypothesis, assumes ν_t to be proportional to the filter width and the resolved shear. Using dimensional analysis, Smagorinski arrived at the following SGS model:

$$\nu_t = C_s^2 \Delta^2 |\bar{S}|, \quad (6.29)$$

where

$$|\bar{S}| = (2\bar{S}_{ij}\bar{S}_{ij})^{\frac{1}{2}} \quad (6.30)$$

is the resolved local strain rate, Δ is the filter width, and C_s is the Smagorinski constant, typically ~ 0.2 .

As stated previously, the Smagorinski model is by far the most commonly used, and has provided reasonable results in countless studies. It is not the general model that is hoped for, though, because adjustments must be made with differing geometries and types of flow. In 1966, Lilly (Lilly, 1995) evaluated C_s analytically in terms of integrals of the velocity correlation function. Assuming a 2nd order central discretization, he obtained

$$C_s \simeq 0.23. \quad (6.31)$$

In 1967, however, Lilly (Lilly, 1967) re-evaluated C_s by assuming the existence of a pre-described inertial range spectrum and then approximately evaluating $|\bar{S}|$ by integrating the spectrum over all resolved wavenumbers. Using this approach he arrived at

$$C_s \simeq 0.18. \quad (6.32)$$

Deardorff published one of the earliest applications of Smagorinski's model to LES in 1970 (Deardorff, 1970). He found that the C_s of Lilly (Lilly, 1995) was too diffusive and concluded that

$$C_s \simeq 0.1 \quad (6.33)$$

was preferred for plane channel flow. In 1980, McMillan et al. (McMillan et al., 1980) found similar results for homogeneous shear flows. In more recent channel flow LES studies (Ferziger and Peric, 1999), where the scales were much better resolved than in Deardorff's study, a value of

$$C_s \simeq 0.065 \quad (6.34)$$

was optimal for the bulk of the flow.

To obtain proper behavior near the wall, rather arbitrary damping of the eddy viscosity must be introduced. The eddy viscosity must also be artificially reduced when stratification and/or rotational effects are present. Very near the wall, C_s should tend toward zero. To accomplish this, a van Driest damping function is often applied:

$$C_s = C_{s_0} \left(1 - e^{-\eta^+/A^+}\right)^2, \quad (6.35)$$

where

$$\eta^+ = \frac{\eta \sqrt{\tau_w/\rho}}{\nu}, \quad (6.36)$$

and η is the normal distance from the wall, τ_w is the wall shear stress, C_{s_0} is the Smagorinski constant away from the wall, and A^+ is a constant $\simeq 25$.

Clark et al. (Clark et al., 1979) studied LES applied to the decay of homogeneous, isotropic turbulence. In order to assess the performance of the Smagorinski model, the velocity field obtained from DNS was filtered. From this, the “exact” SGS stresses were calculated and compared to the modeled ones. It was found that the stresses predicted with the Smagorinski model do not correlate well with the exact. It was found, however, that the volume-averaged SGS dissipation is fairly accurate. McMillan et al. (McMillan et al., 1980) performed a comparable study using homogeneous shear flows and found similar results.

In another set of studies assessing the performance of the Smagorinski model, Piomelli et al. (Piomelli et al., 1990) investigated the model applied to transition in the boundary layer of planar channel flow. It was found that perturbations in the boundary layer were overdamped at the early stages of transition, sometimes leading to relaminarization even at super-critical Reynolds numbers.

Despite the rather long list of problems with the Smagorinski model, it remains popular and is successful at predicting a number of flows. There are two main reasons for this success. The first is that the model does predict the integrated energy transfer from the large scales to the small well, even though there is no means for backscatter effects. The second is that a number of ad hoc corrections have been introduced in the near-wall region and in the transition to turbulence.

6.3.2 Other SGS Models

Though the Smagorinski model is the most commonly used, its many drawbacks have prompted intense research in the area of SGS models. Four of the more popular models will now be discussed. For further reading on the advancement of SGS models, the author recommends the review article by Lesieur and Metais (Lesieur and Metais, 1996).

6.3.2.1 Kraichnan's Spectral Eddy Viscosity Model

Kraichnan's spectral eddy viscosity model (Kraichnan, 1976; Lesieur, 1990) utilizes a sharp Fourier cutoff filter function. The eddy viscosity relation is

$$\nu_t(k, k_c) = 0.441 C_k^{-\frac{3}{2}} \left[\frac{E}{k_c} \right] \nu_t^*, \quad (6.37)$$

where $E = E(k_c)$ is the kinetic energy spectrum at the cutoff frequency k_c , and $\nu_t^* = \nu_t^*(k/k_c)$ is a non-dimensional eddy viscosity. An advantage of this model is that all backscatter effects are included. Though this model is based on an isotropic turbulence assumption, satisfactory results are obtained even if the large scales are neither isotropic nor homogeneous (Metais and Lesieur, 1989; Batchelor et al., 1992). The main drawback to this model is that it is defined in Fourier space, so employing it on complex geometries that necessitate one to work in physical space is difficult.

6.3.2.2 Structure-Function Model

Using the structure-function model presented by Metais and Lesieur (Metais and Lesieur, 1992), one works in physical space with the cutoff frequency defined as $k_c = \pi/\Delta x$. Assuming that k_c is in the inertial region of the Kolmogorov spectrum, from spectral eddy viscosity models Metais and Lesieur arrived at

$$\nu_t(x, \Delta x) = \frac{2}{3} C_k^{-\frac{3}{2}} \left[\frac{E_x}{k_c} \right]^{\frac{1}{2}}, \quad (6.38)$$

where $E_x = E_x(k_c)$ is the local kinetic energy spectrum, and $C_k \simeq 1.5$ is the universal Kolmogorov constant. One advantage of this model is that it takes into account the local intermittency of turbulence and reduces the eddy viscosity in regions where small-scale turbulence has not developed. It works especially well for isotropic turbulence, where it results in a Kolmogorov spectrum for the subgrid scales. Like the Smagorinski model, though, it is too dissipative for low Mach number boundary layer transition in channel flow.

6.3.2.3 Scale Similarity Model

The previously discussed models are all eddy viscosity models, and as such are based on the assumption of a one-to-one correlation between the SGS Reynolds stress and the resolved strain. As mentioned in regard to the Smagorinski model, analysis of DNS and experiments have shown that very little correlation actually exists. The scale similarity model is based on the idea that the important interactions between the subgrid and resolved scales primarily entail the largest eddies of the subgrid scales and the smallest eddies of the resolved scales. Using this idea, Bardina et al. (Bardina et al., 1980) used the small scales of the resolved field to determine the model for the unresolved scales. The SGS Reynolds stress tensor is typically modeled as

$$\tau_{ij} = \bar{\bar{u}_i \bar{u}_j} - \bar{\bar{u}_i} \bar{\bar{u}_j}, \quad (6.39)$$

where the double overbar denotes that the quantity has been filtered twice. This model does not dissipate much energy, consequently it is typically combined with the Smagorinski model in order to be of use. The resulting mixed model has been used with considerable success.

6.3.2.4 Dynamic Models

The final class of models discussed here is termed dynamic models, and was first introduced by Germano et al. (Germano et al., 1991). These models dynamically compute the model coefficients as the simulation progresses based on the energy contained in the smallest resolved scales, rather than imposing them in advance. A double-filtering approach is used to calculate the SGS Reynolds stress tensor. In addition to the grid filter, a test filter of larger width, $\alpha\Delta x$ with $\alpha \geq 1$, is applied. By explicitly calculating the transfers across the test cutoff, an approximation to the transfer across the true cutoff k_c can be made. Though it is wrought with mathematical inconsistencies (Lesieur and Metais, 1996), this approach has produced many results with superior agreement to DNS and experiments. The reasons for this are numerous. The SGS Reynolds stresses given by dynamic models correctly vanish in laminar flows and at solid boundaries. In addition, correct asymptotic behavior in the near wall region is produced. Furthermore, the model provides backscatter mechanisms. Moin et al. (Moin et al., 1991) extended the model to compressible flows, with much success. Zang and Piomelli (Zang and Piomelli, 1993) and Piomelli (Piomelli, 1993) simulated transition in plane channel flow, and produced results that compare very well with DNS. This model has even produced results for rotating turbulence (Squires and Piomelli, 1994) that agree very well with experiments.

6.4 Eddy Viscosity Implementation

6.4.1 Nodal Matrix Equation

Extension of the SFE3D solver to perform LES is rather straightforward. Considering the flowfield variables discussed in Chapter 4 to now be the filtered variables, the addition of two terms,

$$\nu_t \nabla^2 u_i + 2 \frac{\partial \nu_t}{\partial x_j} \overline{S}_{ij}, \quad (6.40)$$

to the right-hand-side of the governing momentum equation is required. These terms are treated in the same fashion as the convective terms. That is, explicit second-order Adams-Bashforth temporal treatment using a pseudo-spectral approach (see Section 4.5). The discretization procedure follows that outlined in Chapter 4. First, the eddy viscosity terms are written in a form that separates the in-plane and transverse components. Utilizing the in-plane gradient operator $\tilde{\nabla}$ introduced previously and tensor notation where the in-plane

coordinate indices are $p = (1, 2)$ and $q = (1, 2)$, (6.40) can be rewritten as

$$\underbrace{\nu_t \tilde{\nabla}^2 u_p + \nu_t \frac{\partial^2 u_p}{\partial z^2}}_{\text{underbrace}} + \frac{\partial \nu_t}{\partial x_q} \left(\frac{\partial u_p}{\partial x_q} + \frac{\partial u_q}{\partial x_p} \right) + \frac{\partial \nu_t}{\partial z} \left(\frac{\partial u_p}{\partial z} + \frac{\partial w}{\partial x_p} \right) \quad (6.41)$$

and

$$\underbrace{\nu_t \tilde{\nabla}^2 w + \nu_t \frac{\partial^2 w}{\partial z^2}}_{\text{underbrace}} + \frac{\partial \nu_t}{\partial x_q} \left(\frac{\partial w}{\partial x_q} + \frac{\partial u_q}{\partial z} \right) + 2 \frac{\partial \nu_t}{\partial z} \frac{\partial w}{\partial z} \quad (6.42)$$

for the in-plane and transverse momentum equations, respectively. Note that the underbraced terms in the above expressions have the same form as the molecular diffusion terms. It is important to note, however, that unlike the molecular viscosity, the eddy viscosity is not constant.

In terms of the nodal matrix equation presented for the laminar solver (4.33), these additional terms result in

$$\begin{bmatrix} \mathcal{K}_{ij} & 0 & 0 & 0 & 0 & 0 & \frac{1}{2}^x q_{ij} & 0 \\ 0 & \mathcal{K}_{ij} & 0 & 0 & 0 & 0 & \frac{1}{2}^x q_{ij} & \\ 0 & 0 & \mathcal{K}_{ij} & 0 & 0 & 0 & \frac{1}{2}^y q_{ij} & 0 \\ 0 & 0 & 0 & \mathcal{K}_{ij} & 0 & 0 & 0 & \frac{1}{2}^y q_{ij} \\ 0 & 0 & 0 & 0 & \mathcal{K}_{ij} & 0 & 0 & A_{2ij} \\ 0 & 0 & 0 & 0 & 0 & \mathcal{K}_{ij} - A_{2ij} & 0 & \\ xx \mathcal{Q}_{ij} & 0 & xy \mathcal{Q}_{ij} & 0 & 0 & A_{2ij} & {}^q S_{ij} & 0 \\ 0 & yx \mathcal{Q}_{ij} & 0 & yy \mathcal{Q}_{ij} & -A_{2ij} & 0 & 0 & {}^q S_{ij} \end{bmatrix} \begin{Bmatrix} \Re(\delta \hat{u}_k)_j \\ \Im(\delta \hat{u}_k)_j \\ \Re(\delta \hat{v}_k)_j \\ \Im(\delta \hat{v}_k)_j \\ \Re(\delta \hat{w}_k)_j \\ \Im(\delta \hat{w}_k)_j \\ \Re(\delta \hat{p}_k)_j \\ \Im(\delta \hat{p}_k)_j \end{Bmatrix} = - \left\{ \vec{\mathcal{D}}_k \right\}_i^n + \frac{3}{2} \left(\left\{ \vec{\mathcal{C}}_k \right\}_i^n + \left\{ \vec{\mathcal{K}}_{ik} \right\}_i^n \right) - \frac{1}{2} \left(\left\{ \vec{\mathcal{C}}_k \right\}_i^{n-1} + \left\{ \vec{\mathcal{K}}_{ik} \right\}_i^{n-1} \right), \quad (6.43)$$

where the previously introduced terms are given in Chapter 4. The new turbulent diffusion load vector is

$$\left\{ \vec{\mathcal{K}}_{ik} \right\}_i = \sum_{n=0}^{N-1} \left\{ \vec{\mathcal{K}}_{in} \right\}_i e^{-\frac{2\pi In}{N} k}, \quad (6.44)$$

where

$$\left\{ \vec{\mathcal{K}}_{in} \right\}_i = \begin{Bmatrix} (\mathcal{K}_{inx})_i \\ (\mathcal{K}_{iny})_i \\ (\mathcal{K}_{inz})_i \\ (\mathcal{K}_{ipn})_i \end{Bmatrix}. \quad (6.45)$$

Performing a SUPG/PSPG semi-discretization (as in Chapter 4) of the new terms (6.40)

gives

$$\begin{aligned}
 (\mathcal{K}_{txn})_i = & - \int_{\Omega_e} \nu_{tn} (\tilde{\nabla} N_i \cdot \tilde{\nabla} u_n) d\Omega_e + \int_{\Omega_e} \nu_{tn} N_i \left(\frac{\partial^2 u}{\partial z^2} \right)_n d\Omega_e + \\
 & \int_{\Omega_e} N_i \left(\frac{\partial \nu_{tn}}{\partial x} \frac{\partial u_n}{\partial x} + \frac{\partial \nu_{tn}}{\partial y} \frac{\partial v_n}{\partial x} \right) d\Omega_e + \\
 & \int_{\Omega_e} N_i \left(\frac{\partial \nu_t}{\partial z} \right)_n \left(\left(\frac{\partial u}{\partial z} \right)_n + \frac{\partial w_n}{\partial x} \right) d\Omega_e + \\
 & \underbrace{\tau_{SUPG} \int_{\Omega_e} (\vec{u}_n \cdot \tilde{\nabla} N_i) \left[\nu_{tn} \left(\tilde{\nabla}^2 u_n + \left(\frac{\partial^2 u}{\partial z^2} \right)_n \right) + \right.} \\
 & \left. 2 \frac{\partial \nu_{tn}}{\partial x} \frac{\partial u_n}{\partial x} + \frac{\partial \nu_{tn}}{\partial y} \left(\frac{\partial u_n}{\partial y} + \frac{\partial v_n}{\partial x} \right) + \left(\frac{\partial \nu_t}{\partial z} \right)_n \left(\left(\frac{\partial u}{\partial z} \right)_n + \frac{\partial w_n}{\partial x} \right) \right] d\Omega_e }_{SUPG} \quad (6.46)
 \end{aligned}$$

$$\begin{aligned}
 (\mathcal{K}_{tyn})_i = & - \int_{\Omega_e} \nu_{tn} (\tilde{\nabla} N_i \cdot \tilde{\nabla} v_n) d\Omega_e + \int_{\Omega_e} \nu_{tn} N_i \left(\frac{\partial^2 v}{\partial z^2} \right)_n d\Omega_e + \\
 & \int_{\Omega_e} N_i \left(\frac{\partial \nu_{tn}}{\partial x} \frac{\partial u_n}{\partial y} + \frac{\partial \nu_{tn}}{\partial y} \frac{\partial v_n}{\partial y} \right) d\Omega_e + \\
 & \int_{\Omega_e} N_i \left(\frac{\partial \nu_t}{\partial z} \right)_n \left(\left(\frac{\partial v}{\partial z} \right)_n + \frac{\partial w_n}{\partial y} \right) d\Omega_e + \\
 & \underbrace{\tau_{SUPG} \int_{\Omega_e} (\vec{u}_n \cdot \tilde{\nabla} N_i) \left[\nu_{tn} \left(\tilde{\nabla}^2 v_n + \left(\frac{\partial^2 v}{\partial z^2} \right)_n \right) + \right.} \\
 & \left. \frac{\partial \nu_{tn}}{\partial x} \left(\frac{\partial v_n}{\partial x} + \frac{\partial u_n}{\partial y} \right) + 2 \frac{\partial \nu_{tn}}{\partial y} \frac{\partial v_n}{\partial y} + \left(\frac{\partial \nu_t}{\partial z} \right)_n \left(\left(\frac{\partial v}{\partial z} \right)_n + \frac{\partial w_n}{\partial y} \right) \right] d\Omega_e }_{SUPG} \quad (6.47)
 \end{aligned}$$

$$\begin{aligned}
 (\mathcal{K}_{tzn})_i = & - \int_{\Omega_e} \nu_{tn} (\tilde{\nabla} N_i \cdot \tilde{\nabla} w_n) d\Omega_e + \int_{\Omega_e} \nu_{tn} N_i \left(\frac{\partial^2 w}{\partial z^2} \right)_n d\Omega_e + \\
 & \int_{\Omega_e} N_i \left(\frac{\partial \nu_{tn}}{\partial x} \left(\frac{\partial u}{\partial z} \right)_n + \frac{\partial \nu_{tn}}{\partial y} \left(\frac{\partial v_n}{\partial z} \right)_n \right) d\Omega_e + \\
 & \int_{\Omega_e} 2 N_i \left(\frac{\partial \nu_t}{\partial z} \right)_n \left(\frac{\partial w_n}{\partial z} \right)_n d\Omega_e + \\
 & \underbrace{\tau_{SUPG} \int_{\Omega_e} (\vec{u}_n \cdot \tilde{\nabla} N_i) \left[\nu_{tn} \left(\tilde{\nabla}^2 w_n + \left(\frac{\partial^2 w}{\partial z^2} \right)_n \right) + \right.} \\
 & \left. \frac{\partial \nu_{tn}}{\partial x} \left(\frac{\partial w_n}{\partial x} + \left(\frac{\partial u}{\partial z} \right)_n \right) + \frac{\partial \nu_{tn}}{\partial y} \left(\frac{\partial w_n}{\partial y} + \left(\frac{\partial v}{\partial z} \right)_n \right) + \right. \\
 & \left. 2 \left(\frac{\partial \nu_t}{\partial z} \right)_n \left(\frac{\partial w}{\partial z} \right)_n \right] d\Omega_e }_{SUPG} \quad (6.48)
 \end{aligned}$$

$$\begin{aligned}
(\mathcal{K}_{tpn})_i = & \underbrace{\tau_{PSPG} \int_{\Omega_e} \tilde{\nabla} N_i \cdot \nu_{tn} \left(\underline{\tilde{\nabla}^2 \vec{u}_n} + \left(\frac{\partial^2 \vec{u}}{\partial z^2} \right)_n \right) d\Omega_e}_{PSPG} + \\
& \underbrace{\tau_{PSPG} \int_{\Omega_e} \frac{\partial N_i}{\partial x} \left[2 \frac{\partial \nu_{tn}}{\partial x} \frac{\partial u_n}{\partial x} + \frac{\partial \nu_{tn}}{\partial y} \left(\frac{\partial u_n}{\partial y} + \frac{\partial v_n}{\partial x} \right) + \right.} \\
& \quad \left. \left(\frac{\partial \nu_t}{\partial z} \right)_n \left(\left(\frac{\partial u}{\partial z} \right)_n + \frac{\partial w_n}{\partial x} \right) \right] d\Omega_e}_{PSPG} + \quad (6.49) \\
& \underbrace{\tau_{PSPG} \int_{\Omega_e} \frac{\partial N_i}{\partial y} \left[\frac{\partial \nu_{tn}}{\partial x} \left(\frac{\partial v_n}{\partial x} + \frac{\partial u_n}{\partial y} \right) + 2 \frac{\partial \nu_{tn}}{\partial y} \frac{\partial v_n}{\partial y} + \right.} \\
& \quad \left. \left(\frac{\partial \nu_t}{\partial z} \right)_n \left(\left(\frac{\partial v}{\partial z} \right)_n + \frac{\partial w_n}{\partial y} \right) \right] d\Omega_e}_{PSPG}
\end{aligned}$$

where \vec{u} is the in-plane velocity vector and $\tilde{\nabla}$ the in-plane gradient operator. In the previous expressions, the SUPG/PSPG terms have been identified via an underbrace and the underlined terms vanish for a P1 element approximation.

As before, the transverse derivatives $(\partial/\partial z)_n$ and $(\partial^2/\partial z^2)_n$ are approximated using second order central finite-differences. The values are calculated via (4.84) and (4.85), respectively, and are then treated as any other nodal DOF in computing the weighted residual form of the eddy viscosity terms.

Finally, the SUPG/PSPG discretization is completed by considering each unknown ϕ as

$$\phi = \sum_j N_j \phi_j. \quad (6.50)$$

The SUPG terms in (6.46)–(6.48) are neglected in this work. The motivation for this is that these terms require a summation over four indices, which is a computationally expensive procedure, especially considering that the terms are small in comparison to the others. The last two integrals in (6.49) are also neglected because they are small yet expensive to compute. Testing via application to the case of flow past a circular cylinder at $Re = 3900$ showed no apparent stability issues as a result of omitting these stabilization terms. The expressions for

the turbulent diffusion terms now become

$$\begin{aligned} (\mathcal{K}_{txn})_i = & -\nu_{tn\ell} u_{nj} \int_{\Omega_e} N_\ell \left(\frac{\partial N_i}{\partial x} \frac{\partial N_j}{\partial x} + \underline{\frac{\partial N_i}{\partial y} \frac{\partial N_j}{\partial y}} \right) d\Omega_e + \\ & \nu_{tn\ell} \left(\frac{\partial^2 u}{\partial z^2} \right)_{nj} \int_{\Omega_e} N_i N_j N_\ell d\Omega_e + \\ & \nu_{tnj} \int_{\Omega_e} N_i \frac{\partial N_\ell}{\partial x} \left(u_{n\ell} \frac{\partial N_j}{\partial x} + v_{n\ell} \frac{\partial N_j}{\partial y} \right) d\Omega_e + \\ & \left(\frac{\partial \nu_t}{\partial z} \right)_{nj} \int_{\Omega_e} N_i N_j \left(N_\ell \left(\frac{\partial u}{\partial z} \right)_{n\ell} + \frac{\partial N_\ell}{\partial x} w_{n\ell} \right) d\Omega_e \end{aligned} \quad (6.51)$$

$$\begin{aligned} (\mathcal{K}_{tyn})_i = & -\nu_{tn\ell} v_{nj} \int_{\Omega_e} N_\ell \left(\frac{\partial N_i}{\partial x} \frac{\partial N_j}{\partial x} + \underline{\frac{\partial N_i}{\partial y} \frac{\partial N_j}{\partial y}} \right) d\Omega_e + \\ & \nu_{tn\ell} \left(\frac{\partial^2 v}{\partial z^2} \right)_{nj} \int_{\Omega_e} N_i N_j N_\ell d\Omega_e + \\ & \nu_{tnj} \int_{\Omega_e} N_i \frac{\partial N_\ell}{\partial y} \left(u_{n\ell} \frac{\partial N_j}{\partial x} + v_{n\ell} \frac{\partial N_j}{\partial y} \right) d\Omega_e + \\ & \left(\frac{\partial \nu_t}{\partial z} \right)_{nj} \int_{\Omega_e} N_i N_j \left(N_\ell \left(\frac{\partial v}{\partial z} \right)_{n\ell} + \frac{\partial N_\ell}{\partial y} w_{n\ell} \right) d\Omega_e \end{aligned} \quad (6.52)$$

$$\begin{aligned} (\mathcal{K}_{tzn})_i = & -\nu_{tn\ell} w_{nj} \int_{\Omega_e} N_\ell \left(\frac{\partial N_i}{\partial x} \frac{\partial N_j}{\partial x} + \underline{\frac{\partial N_i}{\partial y} \frac{\partial N_j}{\partial y}} \right) d\Omega_e + \\ & \nu_{tn\ell} \left(\frac{\partial^2 w}{\partial z^2} \right)_{nj} \int_{\Omega_e} N_i N_j N_\ell d\Omega_e + \\ & \nu_{tnj} \int_{\Omega_e} N_i N_\ell \left(\frac{\partial N_j}{\partial x} \left(\frac{\partial u}{\partial z} \right)_{n\ell} + \frac{\partial N_j}{\partial y} \left(\frac{\partial v}{\partial z} \right)_{n\ell} \right) d\Omega_e + \\ & 2 \left(\frac{\partial \nu_t}{\partial z} \right)_{nj} \left(\frac{\partial w}{\partial z} \right)_{n\ell} \int_{\Omega_e} N_i N_j N_\ell d\Omega_e \end{aligned} \quad (6.53)$$

$$\begin{aligned} (\mathcal{K}_{tpn})_i = & \tau_{PSPG} \nu_{tn\ell} \left[u_{nj} \int_{\Omega_e} \frac{\partial N_i}{\partial x} N_\ell \left(\underline{\frac{\partial^2 N_j}{\partial x^2}} + \underline{\frac{\partial^2 N_j}{\partial y^2}} \right) d\Omega_e + \right. \\ & \left(\frac{\partial^2 u}{\partial z^2} \right)_{nj} \int_{\Omega_e} \frac{\partial N_i}{\partial x} N_j N_\ell d\Omega_e + v_{nj} \int_{\Omega_e} \frac{\partial N_i}{\partial y} N_\ell \left(\underline{\frac{\partial^2 N_j}{\partial x^2}} + \underline{\frac{\partial^2 N_j}{\partial y^2}} \right) d\Omega_e + \\ & \left. \left(\frac{\partial^2 v}{\partial z^2} \right)_{nj} \int_{\Omega_e} \frac{\partial N_i}{\partial y} N_j N_\ell d\Omega_e \right] \end{aligned} \quad (6.54)$$

$\overbrace{\hspace{10em}}$

As before, in the above expressions the underlined terms vanish for P1 element approximations, and repeated subscripts indicate summation.

6.4.2 Analytical Coefficient Evaluation

Because linear triangle elements are used, each of the coefficients appearing in the turbulent diffusion load vector (6.45) can be easily evaluated analytically. After integration of (6.51)–(6.54) one obtains

$$(\mathcal{K}_{txn})_i = \nu_{tn\ell} \left[-u_{nj} \frac{1}{12S_T} ({}^x n_i {}^x n_j + {}^y n_i {}^y n_j) + \frac{S_T}{60} \left(\frac{\partial^2 u}{\partial z^2} \right)_{nj} \max[1, 2(\delta_{ij} + \delta_{i\ell} + \delta_{j\ell})] \right] + \frac{\nu_{tnj}}{12S_T} (u_{n\ell} {}^x n_j + v_{n\ell} {}^y n_j) + \left(\frac{\partial \nu_t}{\partial z} \right)_{nj} \left[\left(\frac{\partial u}{\partial z} \right)_{n\ell} \frac{S_T}{60} \max[1, 2(\delta_{ij} + \delta_{i\ell} + \delta_{j\ell})] + \frac{w_{n\ell} {}^x n_\ell}{24} (1 + \delta_{ij}) \right] \quad (6.55)$$

$$(\mathcal{K}_{tyn})_i = \nu_{tn\ell} \left[-v_{nj} \frac{1}{12S_T} ({}^x n_i {}^x n_j + {}^y n_i {}^y n_j) + \frac{S_T}{60} \left(\frac{\partial^2 v}{\partial z^2} \right)_{nj} \max[1, 2(\delta_{ij} + \delta_{i\ell} + \delta_{j\ell})] \right] + \frac{\nu_{tnj}}{12S_T} (u_{n\ell} {}^x n_j + v_{n\ell} {}^y n_j) + \left(\frac{\partial \nu_t}{\partial z} \right)_{nj} \left[\left(\frac{\partial v}{\partial z} \right)_{n\ell} \frac{S_T}{60} \max[1, 2(\delta_{ij} + \delta_{i\ell} + \delta_{j\ell})] + \frac{w_{n\ell} {}^y n_\ell}{24} (1 + \delta_{ij}) \right] \quad (6.56)$$

$$(\mathcal{K}_{txn})_i = \nu_{tn\ell} \left[-w_{nj} \frac{1}{12S_T} ({}^x n_i {}^x n_j + {}^y n_i {}^y n_j) + \frac{S_T}{60} \left(\frac{\partial^2 w}{\partial z^2} \right)_{nj} \max[1, 2(\delta_{ij} + \delta_{i\ell} + \delta_{j\ell})] \right] + \nu_{tnj} \left(\frac{1 + \delta_{i\ell}}{24} \right) \left({}^x n_j \left(\frac{\partial u}{\partial z} \right)_{n\ell} + {}^y n_j \left(\frac{\partial v}{\partial z} \right)_{n\ell} \right) + \frac{S_T}{30} \left(\frac{\partial \nu_t}{\partial z} \right)_{nj} \left(\frac{\partial w}{\partial z} \right)_{n\ell} \max[1, 2(\delta_{ij} + \delta_{i\ell} + \delta_{j\ell})] \quad (6.57)$$

$$(\mathcal{K}_{tpn})_i = \tau_{PSPG} \nu_{tn\ell} \left(\frac{1 + \delta_{jk}}{24} \right) \left[{}^x n_i \left(\frac{\partial^2 u}{\partial z^2} \right)_{nj} + {}^y n_i \left(\frac{\partial^2 v}{\partial z^2} \right)_{nj} \right] \quad (6.58)$$

6.4.3 Smagorinski Model

The Smagorinski model was discussed in some detail in Section 6.3.1. In this approach, the eddy viscosity at node ℓ of 2D plane n is modeled as

$$\nu_{tn\ell} = C_s^2 \Delta_{n\ell}^2 |\bar{S}|_{n\ell} \quad (6.59)$$

where C_s is the Smagorinski constant, $\Delta_{n\ell}$ is the filter width, and $|\bar{S}|_{n\ell}$ is the strain rate of the resolved scales,

$$|\bar{S}|_{n\ell} = (2\bar{S}_{ij}\bar{S}_{ij})_{n\ell}^{\frac{1}{2}}. \quad (6.60)$$

Expanding the summations in $|\bar{S}|_{n\ell}$ gives

$$\begin{aligned} |\bar{S}|_{n\ell} = & \left[2(\bar{S}_{11}\bar{S}_{11} + \bar{S}_{12}\bar{S}_{12} + \bar{S}_{13}\bar{S}_{13} + \right. \\ & \bar{S}_{21}\bar{S}_{21} + \bar{S}_{22}\bar{S}_{22} + \bar{S}_{23}\bar{S}_{23} + \\ & \left. \bar{S}_{31}\bar{S}_{31} + \bar{S}_{32}\bar{S}_{32} + \bar{S}_{33}\bar{S}_{33}) \right]_{n\ell}^{\frac{1}{2}}. \end{aligned} \quad (6.61)$$

Now, the strain rate tensor is defined as

$$(\bar{S}_{ij})_{n\ell} = \frac{1}{2} \left(\frac{\partial \bar{u}_i}{\partial x_j} + \frac{\partial \bar{u}_j}{\partial x_i} \right)_{n\ell}. \quad (6.62)$$

Substituting this into (6.61) and rearranging gives

$$\begin{aligned} |\bar{S}_{n\ell}| = & \left\{ 2 \left[\left(\frac{\partial \bar{u}_{n\ell}}{\partial x} \right)^2 + \left(\frac{\partial \bar{v}_{n\ell}}{\partial y} \right)^2 + \left(\frac{\partial \bar{w}_{n\ell}}{\partial z} \right)^2 \right] + \right. \\ & \left. \left(\frac{\partial \bar{u}_{n\ell}}{\partial y} + \frac{\partial \bar{v}_{n\ell}}{\partial x} \right)^2 + \left(\frac{\partial \bar{u}_{n\ell}}{\partial z} + \frac{\partial \bar{w}_{n\ell}}{\partial x} \right)^2 + \left(\frac{\partial \bar{v}_{n\ell}}{\partial z} + \frac{\partial \bar{w}_{n\ell}}{\partial y} \right)^2 \right\}^{\frac{1}{2}}. \end{aligned} \quad (6.63)$$

The z -derivatives $(\partial/\partial z)_{n\ell}$ are calculated via the FD expression (4.84). The in-plane derivatives at node ℓ are approximated as the average of the derivatives evaluated within each element associated with the node:

$$\left(\frac{\partial \phi}{\partial x} \right)_\ell = \frac{1}{\underline{\text{EPN}}_\ell} \sum_{m=1}^{\underline{\text{EPN}}_\ell} \phi_j \frac{\partial N_j}{\partial x} \Big|_{elm=\ell} \quad (6.64)$$

$$\left(\frac{\partial \phi}{\partial y} \right)_\ell = \frac{1}{\underline{\text{EPN}}_\ell} \sum_{m=1}^{\underline{\text{EPN}}_\ell} \phi_j \frac{\partial N_j}{\partial y} \Big|_{elm=\ell}, \quad (6.65)$$

where $\underline{\text{EPN}}_\ell$ is the number of elements associated with node ℓ .

For P1 elements,

$$\frac{\partial N_j}{\partial x} = \frac{x_n j}{2S_T} \quad (6.66)$$

$$\frac{\partial N_j}{\partial y} = \frac{y_n j}{2S_T}, \quad (6.67)$$

where ${}^x n_j$ and ${}^y n_j$ are the x - and y -components of the element inward normal vector at side j , respectively. Expressions (6.64) and (6.65) then become

$$\left(\frac{\partial \phi}{\partial x} \right)_e = \frac{1}{\text{EPN}_e} \sum_{m=1}^{\text{EPN}_e} \frac{1}{2S_T} \phi_j {}^x n_j \Big|_{elm=\ell} \quad (6.68)$$

$$\left(\frac{\partial \phi}{\partial y} \right)_e = \frac{1}{\text{EPN}_e} \sum_{m=1}^{\text{EPN}_e} \frac{1}{2S_T} \phi_j {}^y n_j \Big|_{elm=\ell} \quad (6.69)$$

Deardorff (Deardorff, 1970), Moin and Kim (Moin and Kim, 1982), and others suggest the filter width Δ be the characteristic length

$$\Delta = (\Delta_1 \Delta_2 \Delta_3)^{\frac{1}{3}}, \quad (6.70)$$

where Δ_i is the element length in the i direction. The triangular prism elements used in this solver do not lend themselves well to this definition. Taking inspiration from (6.70), however, in this work the filter width Δ is taken to be

$$\Delta = V_e^{\frac{1}{3}}, \quad (6.71)$$

where V_e is the volume of the element.

6.5 Near-Wall Treatment

Wall functions are not implemented into the SFELES solver. Instead, the grid must be refined sufficiently to resolve the near-wall gradients. This requires that the first grid point be located at a distance $y^+ < 2$ from the wall, and the streamwise and lateral grid spacing be of order $\Delta x^+ \simeq 50 - 150$ and $\Delta z^+ \simeq 15 - 40$, respectively. In the previous expressions, the distances are given in viscous wall units:

$$\eta^+ = \frac{\eta \sqrt{\tau_w / \rho}}{\nu}, \quad (6.72)$$

where τ_w is the shear stress at the wall. Certainly then, this approach cannot be used at extremely high Reynolds numbers because as the Reynolds number is increased, an increasing number of grid points must be employed.

As mentioned in Section 6.3.1, the Smagorinski constant must be reduced near solid walls. A van Driest damping function is used in SFELES to accomplish this. That is, the Smagorinski model (6.29) becomes

$$\nu_t = (C_s f_w)^2 \Delta^2 |\bar{S}|, \quad (6.73)$$

where the damping function is

$$f_w = 1 - e^{-y^+ / A^+} \quad (6.74)$$

and y^+ is the normal distance to the nearest solid wall and $A^+ = 25$.

Evaluating the distance between a node and the nearest wall is accomplished by the straightforward approach of looping over all boundary segments for each node and determining the smallest distance. This is a somewhat expensive process, but it need only be performed once at the beginning of a simulation.

6.6 Aliasing Errors

When a pseudo-spectral approach is used for the nonlinear terms, aliasing errors are introduced, especially in convection-dominated flows. These errors occur any time nonlinear terms are evaluated approximately in physical space. For instance, consider the 1D case

$$\hat{s}(k) = \hat{a}(k)\hat{b}(k), \quad (6.75)$$

where

$$\hat{a}(k) = \mathcal{F}[a(x)] \quad (6.76)$$

$$\hat{b}(k) = \mathcal{F}[b(x)]. \quad (6.77)$$

If a and b have harmonic components up to N , where N is the number of Fourier modes in the DFT, the product ab has harmonic components up to $2N$. The aliasing errors come about because the DFT with N points cannot distinguish between wavenumbers that are multiples of N .

Much effort has been expended in determining ways to control or remove aliasing errors in pseudo-spectral approaches to solving the NS equations (Canuto et al., 1988; Orzag, 1971b; Patterson and Orzag, 1971). The process of removing aliasing errors is termed dealiasing. The approach that has become standard in spectral methods for DNS and LES is $\frac{3}{2}$ -dealiasing. The key to this technique is to use a DFT with at least $\frac{3}{2}N$ points, where N is again the number of Fourier modes. That is, because the nonlinear interactions generate modes up to double the original cutoff frequency, aliasing can be eliminated by evaluating the convective terms at more nodes and only keeping the lowest N modes from the DFT. It has been shown (Orzag, 1971b) that a minimum of $\frac{3}{2}N$ nodes is required for dealiasing. This approach comes at a substantial cost because the evaluation of the convective terms, which can take up to 25–30% of the overall computation time, is increased by 50%. In addition, the number of discrete points seen by the DFT is no longer a power of two. Although fast DFTs exist for this case, they are not as efficient as those developed for 2^n points (Press et al., 1992).

Dealiasing can be avoided if the aliasing errors are small compared to the truncation errors and the SGS terms. For the divergence and convective forms of the convective terms, aliased spectral methods are not energy conserving and can be unstable. It has been shown, however, that if the skew-symmetric or rotational forms are used, spectral methods are energy

conserving even in the presence of aliasing errors (Zang, 1991; Kravchenko and Moin, 1997). It has also been shown that when the skew-symmetric form is used, the results are well-behaved without dealiasing and that the aliasing errors, especially at the higher wavenumbers, are significantly reduced (Kravchenko and Moin, 1997).

To address aliasing errors in the SFELES solver, the skew-symmetric form of the convective terms is used. This form is an average of the convective and divergence forms, or

$$\frac{1}{2}[\nabla \cdot (\vec{u}\vec{u}) + \vec{u} \cdot \nabla \vec{u}]. \quad (6.78)$$

Rearranging this expression gives

$$\vec{u} \cdot \nabla \vec{u} + \frac{1}{2}\vec{u}(\nabla \cdot \vec{u}). \quad (6.79)$$

The first term in the above expression is the convective form used in the previous discussions of the FE/spectral discretization. Implementing the skew-symmetric form simply requires the term

$$\frac{1}{2}\vec{u}(\nabla \cdot \vec{u}) \quad (6.80)$$

to be added to the convective load vector $\{\vec{\mathcal{C}}_k\}_i$ in (6.43). In physical space, the following terms are added to $\{\vec{\mathcal{C}}_n\}_i$ in (4.91)

$$(ss_{xn})_i = \frac{1 + \delta_{i\ell}}{48} u_\ell (u_j^x n_j + v_j^y n_j) + \frac{S_T}{120} u_\ell \left(\frac{\partial w}{\partial z} \right)_j \max[1, 2(\delta_{ij} + \delta_{i\ell} + \delta_{j\ell})] \quad (6.81)$$

$$(ss_{yn})_i = \frac{1 + \delta_{i\ell}}{48} v_\ell (u_j^x n_j + v_j^y n_j) + \frac{S_T}{120} v_\ell \left(\frac{\partial w}{\partial z} \right)_j \max[1, 2(\delta_{ij} + \delta_{i\ell} + \delta_{j\ell})] \quad (6.82)$$

$$(ss_{zn})_i = \frac{1 + \delta_{i\ell}}{48} w_\ell (u_j^x n_j + v_j^y n_j) + \frac{S_T}{120} w_\ell \left(\frac{\partial w}{\partial z} \right)_j \max[1, 2(\delta_{ij} + \delta_{i\ell} + \delta_{j\ell})] \quad (6.83)$$

$$(ss_{pn})_i = 0. \quad (6.84)$$

SUPG/PSPG stabilization is neglected for the skew-symmetric terms because it requires a summation over four indices, which is a computationally expensive procedure considering that the terms are very small in comparison to the others.

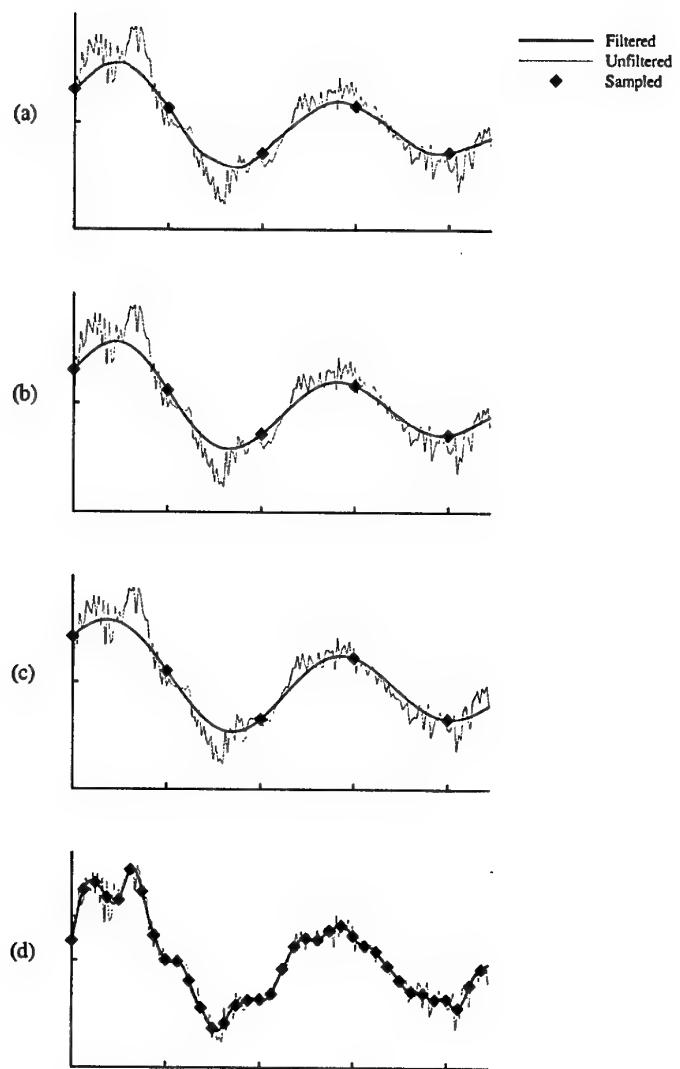


Fig. 6.3 – Filtering of a 1D function. a) tophat filter function, b) Gaussian filter function, c) sharp Fourier cutoff filter function, and d) sharp Fourier cutoff filter function with the filter width reduced by a factor of eight.

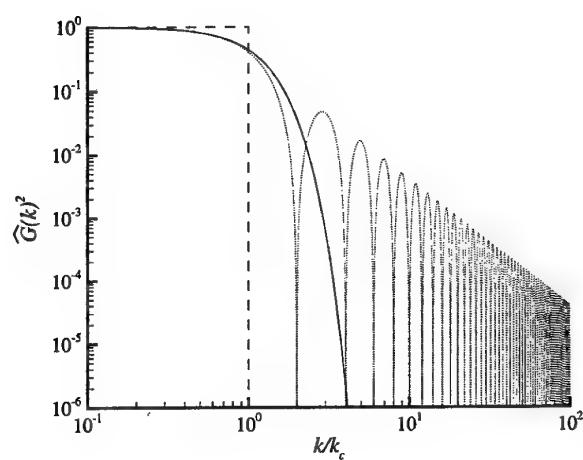


Fig. 6.4 – Attenuation factors of the three most common filter functions. ····· tophat, — Gaussian, - - - sharp Fourier cutoff.

7. LARGE-EDDY SIMULATION TEST CASES

A validation test case of flow past a circular cylinder at $Re = 3900$ is presented in this chapter. Results in terms of qualitative behavior and quantitative velocity and turbulence values are analyzed. The SFELES solver is then used to investigate the effects of a wake splitter plate attached to a cylinder. Results are presented for multiple splitter plate lengths at $Re = 3900$.

7.1 Circular Cylinder

7.1.1 Problem Description

Discussions of laminar circular cylinder flow have been presented in Sections 3.3 and 5.1. According to Prasad and Williamson (Prasad and Williamson, 1997), the onset of turbulence occurs at $Re \approx 1200$; it is at this Reynolds number that the shear layers become unstable. However, there is no firm consensus on the value of this critical Reynolds number—other studies have stated it to be anywhere from 300 to 3000. In this study, $Re = 3900$ is chosen because it is definitely above the critical Reynolds number and yet low enough such that an exorbitant number of nodes are not required near the solid walls. In addition, this has been a popular Reynolds number for studies of this kind, and PIV and hot wire data is available from Lourenco and Shih (Lourenco and Shih, 1993) and Ong and Wallace (Ong and Wallace, 1996), respectively. Many numerical simulations have also been performed at this Reynolds number. These include Beudan and Moin (Beudan and Moin, 1994), who used a high-order upwind FD scheme, Mittal and Moin (Mittal and Moin, 1997), who used a combined 2nd-order central FD/spectral scheme, and Kravchenko and Moin (Kravchenko and Moin, 1998), using a B-spline method. A number of important pieces of information have been learned from these studies. The first is that energy-conserving schemes are more suitable for these simulations. Also, the dissipation due to truncation errors can become more substantial than that of the SGS model in some situations when using lower-order FD schemes. Though the numerical simulations produced results in good agreement with the experiments in the near wake ($x/D < 4$), excessive numerical dissipation caused under-prediction of the turbulent fluctuations farther downstream. This numerical dissipation was most pronounced in the upwind FD simulations.

This test case serves to validate the SFELES solver, at least at this relatively low Reynolds number. The primary goal is not to provide new information regarding the physics of the flow, but rather to present the results of this work, both of qualitative and quantitative nature, in comparison to published values.

7.1.2 Computational Mesh

A perspective view of the computational mesh used for this study is shown in Figure 7.1. A Dirichlet velocity inlet condition is applied $5D$ upstream of the cylinder center, while a Neumann outlet condition exists $20D$ downstream. Slip-wall conditions are applied at the lateral extents of the domain, located $7D$ from the cylinder center. As discussed in Section 5.2, a shedding frequency slightly higher than in freestream flow is expected due to the proximity of these lateral walls. The spanwise dimension is $L_z/D = \pi$. This is the same dimension used in the numerical simulations mentioned in the previous section (Beudan and Moin, 1994; Mitall and Moin, 1997; Kravchenko and Moin, 1998) and was found in these studies to be sufficiently large.

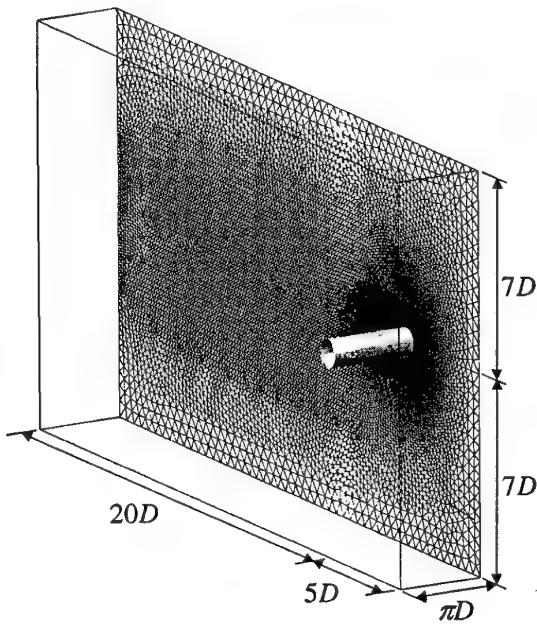


Fig. 7.1 – Turbulent circular cylinder flow: overview of the computational domain and mesh containing 23,500 in-plane triangle elements (12,000 in-plane nodes) and 32 Fourier modes.

In the 2D plane the mesh consists of 23,500 triangle elements with approximately 12,000 nodes. As seen in Figure 7.1, nodes are clustered in the wake region, with characteristic element side lengths of $0.10D$ in the near wake and $0.19D$ farther downstream. Outside the wake region the mesh is more coarse, with the characteristic element side length being $0.5D$ at the upstream and lateral domain extents. Elements are also clustered toward the cylinder in order to better capture the initial development of the turbulent wake, as illustrated in Figure 7.2. Boundary layer wedge-type elements are used very near the cylinder walls, with the perpendicular dimension of the first element off the cylinder being $0.0025D$. A posteriori

analysis shows the first node being located at $y^+ = 1.8$, and around 5 nodes inside the boundary layer at $\theta = 45^\circ$ (with 0° being at $y = 0$ on the upstream side of the cylinder). There exist 110 nodes on the cylinder wall, resulting in an element side length of $0.0275D$. The mesh in this region is shown in Figure 7.3(a).

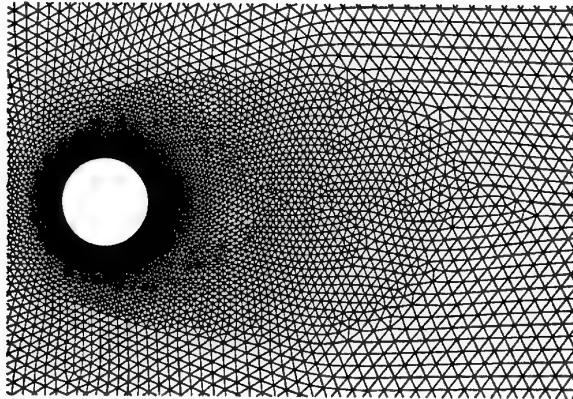


Fig. 7.2 – Turbulent circular cylinder flow: view of the 2D mesh in the near-wake region.

According to the studies by Mansy et al (Mansy et al., 1994) and Williamson et al (Williamson et al., 1995), the spanwise wavelength of the fluid structures near the cylinder scale as

$$\lambda_z/D = \alpha Re^{-1/2}, \quad (7.1)$$

where $\alpha \approx 20-25$. At $Re = 3900$ the structures are expected to have wavelengths $\lambda_z/D \approx 0.4$. Using a spectral discretization in the transverse direction, at least two points are required per wavelength. Consequently, with $L_z/D = \pi$ a minimum of 16 transverse points are required. In this study, 32 points were used in order to ensure sufficient resolution. The overall mesh contains approximately 380,000 nodes.

A refined mesh was also generated to verify sufficient spatial resolution. This mesh contains 43,000 in-plane elements with approximately 22,000 nodes. Characteristic element side lengths in the wake region range from $0.071D$ near the cylinder to $0.22D$ near the outlet. The perpendicular dimension of the first wedge-type element off the cylinder is $0.0015D$, and the typical element side length along the cylinder wall is $0.017D$. A posteriori analysis shows the first node off the cylinder being located at $y^+ = 1.0$ and about 8 nodes inside the boundary layer at $\theta = 45^\circ$. The mesh in this region is shown in Figure 7.3(b). No refinement was made in the transverse direction because 32 planes are already sufficient to resolve wavelengths half the size expected. The refined mesh contains approximately 700,000 nodes.

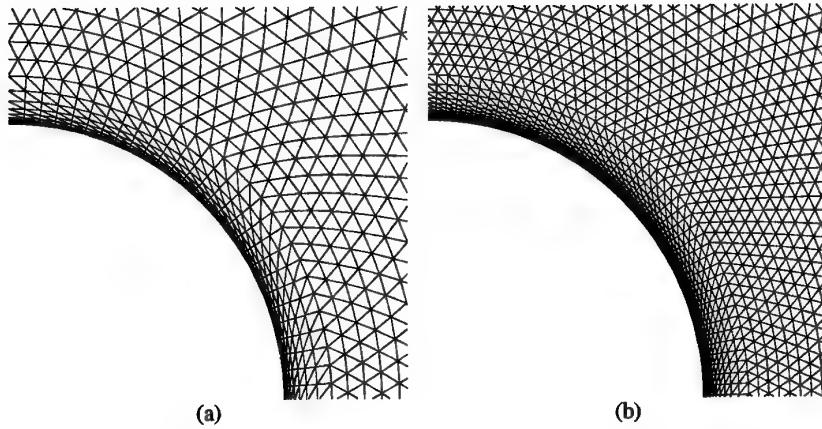


Fig. 7.3 – Turbulent circular cylinder flow: view of the 2D mesh very near the cylinder, showing the wedge-type boundary layer elements for the (a) coarse and (b) fine meshes.

7.1.3 Results

Both the coarse and fine mesh simulations were initialized using potential flow. To decrease computation time, the solution was allowed to develop as a 2D flow until the Karman vortex street was fully developed. To hasten the development of asymmetric shedding, a time-dependent rotation was applied to the cylinder at the beginning of the simulation for a short time; first in one direction and then the opposite. After the 2D Karman vortex street was sufficiently developed, a small perturbation was applied at the inlet boundary for a short time to initiate the 3D features of the flow. The simulation was then allowed to advance until a statistically steady vortex shedding developed. Finally, the simulation was run an additional six shedding cycles, $TU_\infty/D \approx 30$, to allow calculation of the flow statistics. The time step throughout the simulation was fixed at $\Delta t U_\infty/D = 0.001$. No turbulent fluctuations are introduced at the inlet. Because $Re = 3900$ is in the subcritical Reynolds number regime ($Re \lesssim 100,000$), the free-stream flow and cylinder boundary layers are laminar—turbulence develops only in the shear layers and wake region.

An instantaneous perspective view of the resulting flowfield is shown in Figure 7.4 via surfaces of constant vorticity magnitude. A closer view of the vorticity iso-surfaces is shown in Figure 7.5. The long shear layers attached to the cylinder are seen to roll up into the common Karman vortex street. The length of the recirculation region behind the cylinder is predicted to extend approximately $1D$ downstream of the cylinder, which is in agreement with experimental observations (Williamson, 1996b). The surfaces in Figures 7.4 and 7.5 are colored according to velocity magnitude, with blue representing low values and red representing high values. The fluid inside the recirculation region is slow moving, but contains small

scale turbulent structures. Figure 7.6 shows a bottom view of the vorticity magnitude iso-surfaces. In this figure one sees the separating shear layers roll into vortices with diameters on the order of $1D$, which is confirmed in experimental observations by Chyu and Rockwell (Chyu and Rockwell, 1996) and Prasad and Williamson (Prasad and Williamson, 1997) at this Reynolds number.



Fig. 7.4 – Turbulent circular cylinder flow: perspective view of instantaneous surfaces of constant vorticity magnitude, $\omega D/U_\infty = 2.5$. The surfaces are colored according to velocity magnitude, with blue representing low values and red representing high values.

Figures 7.7–7.9 show contours of constant streamwise, lateral, and transverse velocity, respectively, on the mid-plane of the channel ($y = 0$ plane). From the streamwise velocity contours, the extent of the recirculation region is easily seen, along with the high levels of turbulent fluctuations inside the region. Figure 7.8 shows alternating regions of positive and negative lateral velocity in the streamwise direction, highlighting the Karman vortex street. These figures, particularly the transverse velocity contours, show that small scale structures are dominant near the cylinder, with larger scales dominating downstream. According to Williamson et al (Williamson et al., 1995) and Chyu and Rockwell (Chyu and Rockwell, 1996), the fluid structures far downstream of the cylinder have experimentally observed wavelengths

$$\lambda_z/D \sim 1. \quad (7.2)$$

The SFELES solution agrees well with this observation. It is important to notice that small-scale fluctuations do exist far downstream in Figures 7.7–7.9, though they are dominated by the large structures. The upwind FD simulations of Beudan and Moin (Beudan and Moin, 1994) show no small-scale turbulence in the far wake, though other numerical and



Fig. 7.5 – Turbulent circular cylinder flow: front view of instantaneous surfaces of constant vorticity magnitude, $\omega D/U_\infty = 2.5$, in the region $-0.5D \leq x \leq 10D$. The surfaces are colored according to velocity magnitude, with blue representing low values and red representing high values.

experimental studies do. This emphasizes that the SUPG formulation does not suffer from the overly-diffusive nature of typical upwind schemes.

The final qualitative results presented illustrate the development of the alternating vortex shedding phenomenon. Figure 7.10 shows vorticity magnitude iso-surfaces in the very near wake at five instants during the shedding cycle. In the first frame the bottom vortex, which is completely rolled up, is beginning to wash downstream. At the same instant, the top vortex is already formed and in the process of rolling up. In the second and third frames the top vortex grows in size, which draws fluid into the wake from the opposite side. Some of this fluid is entrained into the top vortex, while some replenishes the lost fluid in the recirculation region. By the fourth frame the top vortex is completely rolled up and is being carried downstream, while the development of the bottom vortex is well underway. Finally, in the fifth frame the bottom vortex is nearly rolled up and the process repeats. These results agree with the observations of Gerrard (Gerrard, 1966) and Perry et al (Perry et al., 1982) regarding the physical mechanics of the formation region behind bluff bodies.

As mentioned previously, statistical information was gathered over six shedding cycles. Table 7.1 shows quantitative results for mean drag coefficient (\bar{C}_{D_P}), minimum mean streamwise velocity (\bar{U}_{min}), formation length of the recirculation region (L_f), location of the shear layer separation points (θ_{sep}), and non-dimensional shedding frequency (St). These results are compared with available experimental measurements. With the exception of C_{D_P} , the



Fig. 7.6 – Turbulent circular cylinder flow: bottom view of instantaneous surfaces of constant vorticity magnitude, $\omega D/U_\infty = 2.5$, in the region $-0.5D \leq x \leq 10D$.



Fig. 7.7 – Turbulent circular cylinder flow: instantaneous streamwise velocity contours on the $y = 0$ plane and in the region $-0.5D \leq x \leq 10D$. Shown are 16 contours from $-1.2 \leq u \leq 1.2$, with the $u = 0$ contour highlighted.

SFELES results are within the experimental error bounds.

Table 7.1 – Turbulent circular cylinder flow: comparison of flow parameters from SFELES at $Re = 3900$ with various experiments. C_{D_p} is from Norberg ($Re = 4020$) (Norberg, 1993), \bar{U}_{min} from Lorenc and Shih (Lourenco and Shih, 1993), L_f/D from Cardell (Cardell, 1993), θ_{sep} from Son and Hanratty ($Re = 5000$) (Son and Hanratty, 1969), and St from Ong and Wallace (Ong and Wallace, 1996). Unless stated otherwise, the above experimental values were measured at $Re = 3900$.

	C_{D_p}	\bar{U}_{min}	L_f/D	θ_{sep}	St
SFELES	1.09	-0.29	1.30	88.0°	0.2179
Experiment	$0.99 \pm .05$	$-0.24 \pm .1$	$1.4 \pm .1$	$86^\circ \pm 2^\circ$	$0.215 \pm .005$

More statistical information, in terms of mean velocities and turbulence statistics, is compared with experimental and numerical studies on the following pages. The experimental

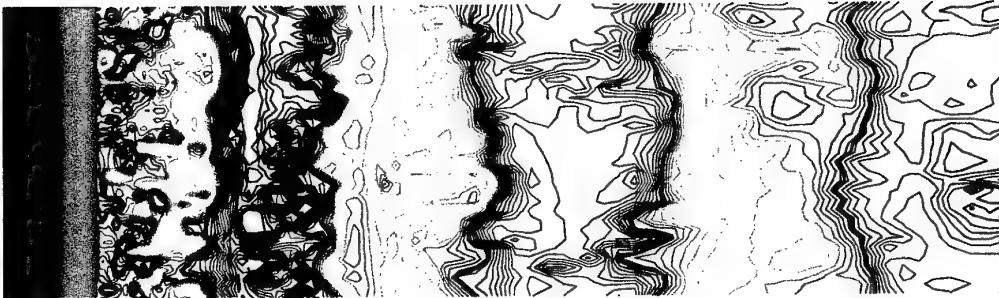


Fig. 7.8 – Turbulent circular cylinder flow: instantaneous lateral velocity contours on the $y = 0$ plane and in the region $-0.5D \leq x \leq 10D$. Shown are 16 contours from $-1.0 \leq v \leq 1.0$, with the $v = 0$ contour highlighted.

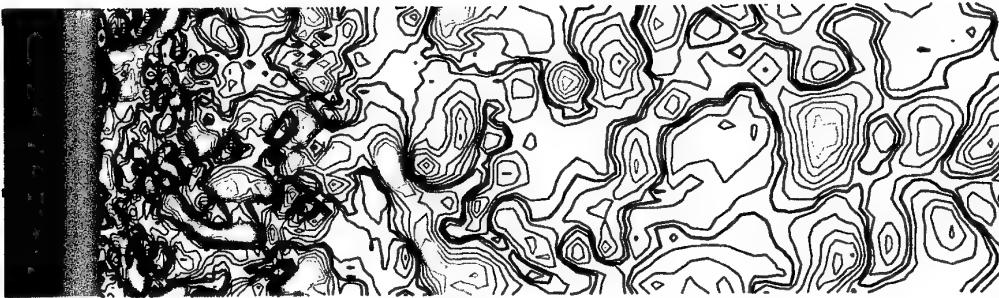


Fig. 7.9 – Turbulent circular cylinder flow: instantaneous transverse velocity contours on the $y = 0$ plane and in the region $-0.5D \leq x \leq 10D$. Shown are 16 contours from $-1.0 \leq w \leq 1.0$, with the $w = 0$ contour highlighted.

studies entail the PIV measurements of Lourenco and Shih (Lourenco and Shih, 1993) and the hot wire measurements of Ong and Wallace (Ong and Wallace, 1996). The numerical study used for comparison is that of Kravchenko and Moin (Kravchenko and Moin, 1998), where a B-spline method was employed. This study is chosen because the results are in good agreement with experimental measurements as well as the numerical results of Beudan and Moin (Beudan and Moin, 1994) and Mittal and Moin (Mittal and Balachandar, 1997) (at least in the near wake, $x/D < 4$), where second-order upwind and central FD methods were used, respectively. The results of Kravchenko and Moin (Kravchenko and Moin, 1998) were achieved on a mesh with approximately 1.3 million nodes—nearly double the fine-mesh used in this work—and extended $30D$ from the cylinder in the 2D plane and had a spanwise dimension of πD . SFELES results are presented for both the coarse and fine meshes. The profiles used for comparison were digitized from published plots, consequently an error estimated at $\pm 5\%$ is introduced.

Figure 7.11 shows the mean streamwise velocity along the $y = 0$ line in the wake up to $x/D = 10$. The SFELES results show better agreement with the PIV measurements than do the B-spline results in terms of the size and strength of the recirculation region. In the region $2.5 \leq x \leq 4$, the SFELES results do not agree with the PIV. However, the PIV results (Lourenco and Shih, 1993) are the only to show a dip in the streamwise velocity in this region.

Mean streamwise and lateral velocity profiles at three streamwise locations in the wake ($x/D = 1.54$, 3.00, and 5.00) are shown in Figures 7.12 and 7.13, respectively. The results are in reasonable agreement, with the largest discrepancies occurring at the station inside the recirculation region at $x/D = 1.54$. Finally, turbulence statistics profiles are shown at the same three streamwise stations in Figures 7.14–7.16. Again, the SFELES results are in agreement with the published values. In fact, the SFE3D solver generally produces results in better agreement with the experiments than the B-spline solver (especially in the very near wake).

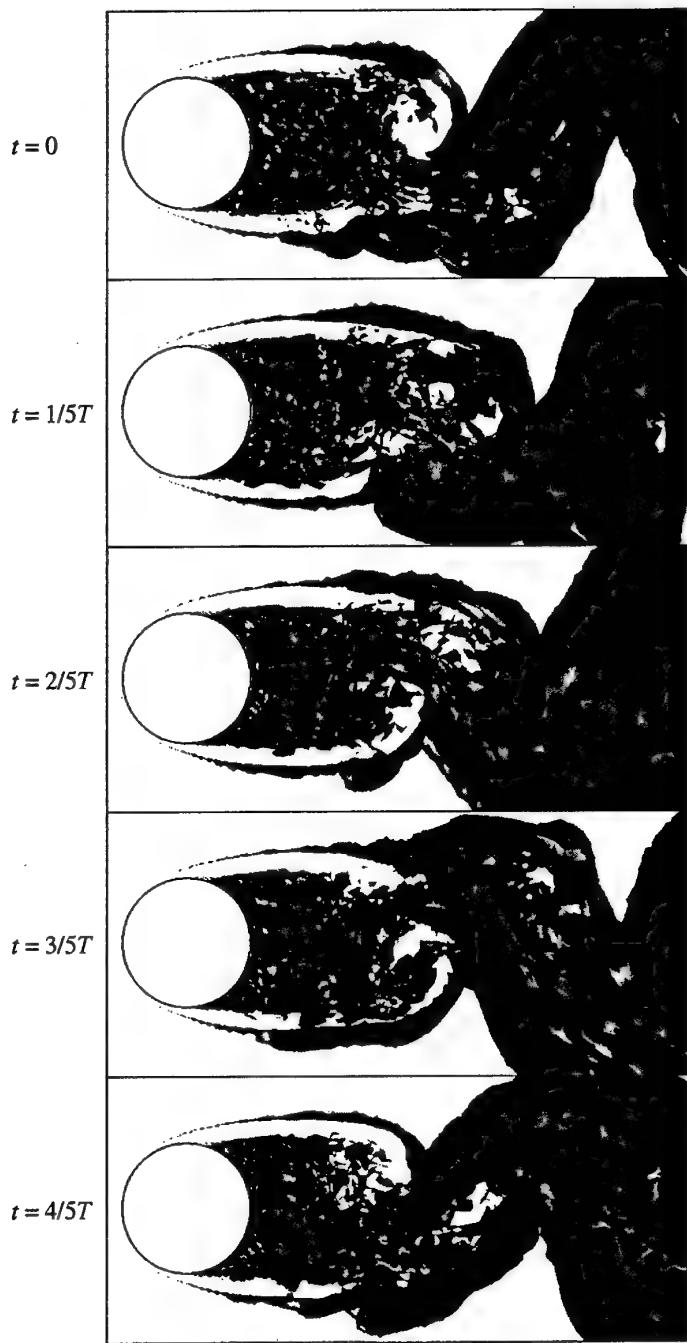


Fig. 7.10 – Turbulent circular cylinder flow: instantaneous surfaces of constant vorticity magnitude, $\omega D/U_\infty = 2.5$, in the very near wake at five instants during the shedding cycle. T is the time interval of one complete shedding cycle.

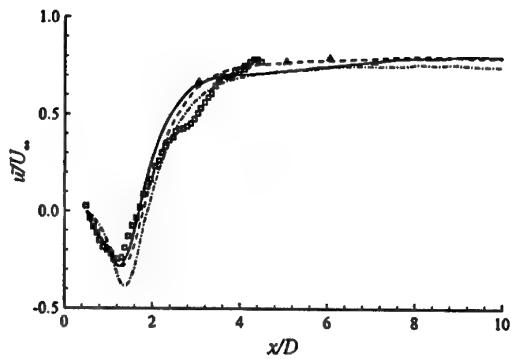


Fig. 7.11 – Turbulent circular cylinder flow: mean streamwise velocity along the line $y = 0$ at $Re = 3900$. — SFELES fine grid, ----- SFELES coarse grid, - - - B-spline simulations by Kravchenko and Moin (Kravchenko and Moin, 1998), □ PIV measurements of Lourenco and Shih (Lourenco and Shih, 1993), △ hot wire measurements of Ong and Wallace (Ong and Wallace, 1996).

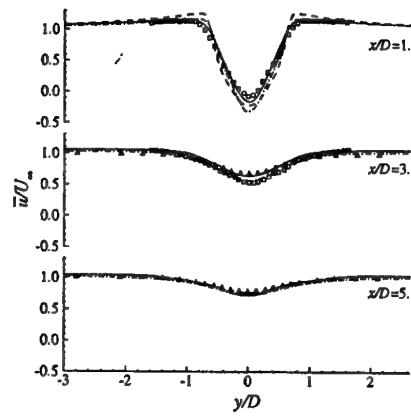


Fig. 7.12 – Turbulent circular cylinder flow: mean streamwise velocity at different streamwise locations in the wake at $Re = 3900$. — SFELES fine grid, ----- SFELES coarse grid, - - - B-spline simulations by Kravchenko and Moin (Kravchenko and Moin, 1998), □ PIV measurements of Lourenco and Shih (Lourenco and Shih, 1993), △ hot wire measurements of Ong and Wallace (Ong and Wallace, 1996).

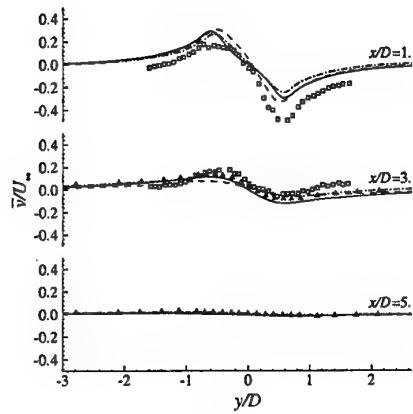


Fig. 7.13 – Turbulent circular cylinder flow: mean lateral velocity at different streamwise locations in the wake at $Re = 3900$. — SFELES fine grid, - - - SFELES coarse grid, - · - B-spline simulations by Kravchenko and Moin (Kravchenko and Moin, 1998), □ PIV measurements of Lourenco and Shih (Lourenco and Shih, 1993), △ hot wire measurements of Ong and Wallace (Ong and Wallace, 1996).

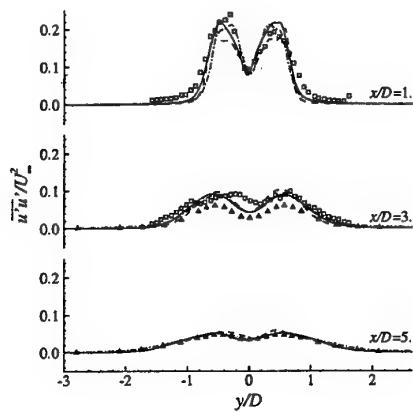


Fig. 7.14 – Turbulent circular cylinder flow: time-averaged streamwise velocity fluctuations at different streamwise locations in the wake at $Re = 3900$. — SFELES fine grid, - - - SFELES coarse grid, - · - B-spline simulations by Kravchenko and Moin (Kravchenko and Moin, 1998), □ PIV measurements of Lourenco and Shih (Lourenco and Shih, 1993), △ hot wire measurements of Ong and Wallace (Ong and Wallace, 1996).

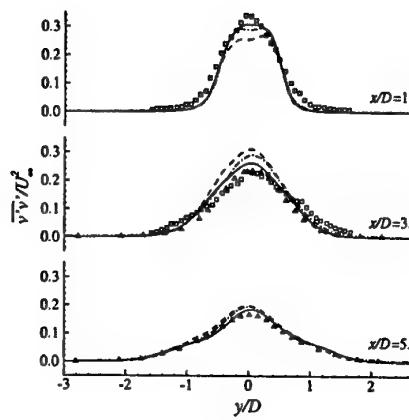


Fig. 7.15 – Turbulent circular cylinder flow: time-averaged lateral velocity fluctuations at different streamwise locations in the wake at $Re = 3900$. — SFELES fine grid, ----- SFELES coarse grid, - - - B-spline simulations by Kravchenko and Moin (Kravchenko and Moin, 1998), \square PIV measurements of Lourenco and Shih (Lourenco and Shih, 1993), \triangle hot wire measurements of Ong and Wallace (Ong and Wallace, 1996).

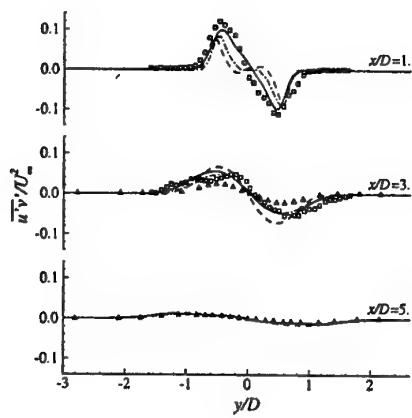


Fig. 7.16 – Turbulent circular cylinder flow: time-averaged Reynolds shear stress at different streamwise locations in the wake at $Re = 3900$. — SFELES fine grid, ----- SFELES coarse grid, - - - B-spline simulations by Kravchenko and Moin (Kravchenko and Moin, 1998), \square PIV measurements of Lourenco and Shih (Lourenco and Shih, 1993), \triangle hot wire measurements of Ong and Wallace (Ong and Wallace, 1996).

7.2 Circular Cylinder With Wake Splitter Plate

7.2.1 Problem Description

A splitter plate attached to the downstream portion of a cylinder has long been used as a means of passive control of the wake flow characteristics. Roshko (Roshko, 1954) was the first to investigate this configuration. His experiments at $Re = 1.45 \times 10^4$ showed that a splitter plate with length $L/D = 1.54$ caused a reduction in shedding frequency. It was also seen that with a splitter plate of $L/D = 5$ the vortex shedding was completely suppressed and the pressure drag was reduced to 63% of the no splitter plate case. Gerrard (Gerrard, 1966) increased the knowledge base by investigating circular cylinders with splitter plates at $Re = 2.0 \times 10^4$. The Strouhal number was found to decrease for splitter plate lengths $0 \leq L/D \leq 1$, and then increase until $L/D = 2$, the maximum length investigated. Even small splitter plates, $L/D = 0.125$, reduced the pressure drag to 83% of the no splitter plate case, and a maximum drag reduction to 68% was achieved at $L/D = 1$.

Apelt et al (Apelt et al., 1973) and Apelt and West (Apelt and West, 1975) made significant contributions in their investigation at $Re = 10^4\text{--}5 \times 10^4$ and splitter plate lengths up to $L/D = 7$. It was found that the addition of a splitter plate to the circular cylinder reduced the pressure drag significantly by stabilizing the shear layer separation points and producing a narrower wake. In addition, the base pressure was raised by as much as 50% and the Strouhal number was affected to a lesser degree. The maximum change in drag coefficient and wake width occurred at $L/D \approx 1$, agreeing with the observations of Gerrard (Gerrard, 1966). As with the studies of Roshko (Roshko, 1954), at $L/D > 5$ the vortex shedding was completely suppressed. It was also noted that the position of vortex formation moves downstream as L/D increases from 0 to 1, with the location being very near the end of the plate. At larger L/D , the formation position remains nearly the same with respect to the cylinder, and moves slightly away from the plate in the lateral direction.

Anderson and Szewczyk (Anderson and Szewczyk, 1997) investigated straight and sinusoidal splitter plate geometries using hot wire measurements and flow visualization. This study, at $Re = 3.5 \times 10^4\text{--}4.6 \times 10^4$ shows a marked increase in Strouhal number for very small splitter plate lengths, $L/D \leq 0.25$, followed by a decrease for $0.25 \leq L/D \leq 0.8$, and an increase up to $L/D \leq 1.5$. In addition, a drag reduction was measured with increasing splitter plate length up to $L/D \approx 1$, again highlighting the significance of the $L/D = 1$ case. In this study a significant decrease in separation point movement was seen even for very small L/D , supporting the investigation of Apelt et al (Apelt et al., 1973). Figure 7.17 from Anderson and Szewczyk (Anderson and Szewczyk, 1997) summarizes the four splitter plate regions that have been identified and are described below. Frame (a) shows the no splitter plate ($L/D = 0$) case, where the shear layers move freely in phase with the vortex shedding. In this case the Strouhal number is Reynolds number independent at $St \approx 0.20$. Frame (b) shows Region I, the Stabilizing Region ($L/D \leq 0.25$). The presence of the splitter

plate stabilizes the separation points, and a progressive increase in Strouhal number is seen with L/D . The strength of the shear layers is increased in comparison to the $L/D = 0$ case, and consequently their interaction is increased in the near wake. Anderson and Szewczyk (Anderson and Szewczyk, 1997) term Region II, shown in frame (c), the Elongation Region ($0.25 \leq L/D \leq 0.75$). This region sees a progressive decrease in Strouhal number with L/D . It is characterized by a large shear layer interaction length, which gives rise to increased entrainment that reduces the shear layer interaction. Consequently, a progressive increase in formation length is seen with L/D . Region III, in frame (d) is termed the Reduced Entrainment Region ($0.75 \leq L/D \leq 1.5$). In this region the formation length remains constant, the interaction length decreases, and a progressive increase in Strouhal number is seen with L/D . The final region, the Splitter Plate-Vortex Interaction Region ($L/D \geq 1.5$) is shown in frame (e). The shear layer interaction is very small in this region and the Strouhal number increases with L/D .

Other experimental studies, for instance (Unal and Rockwell, 1987; Hasan and Budair,

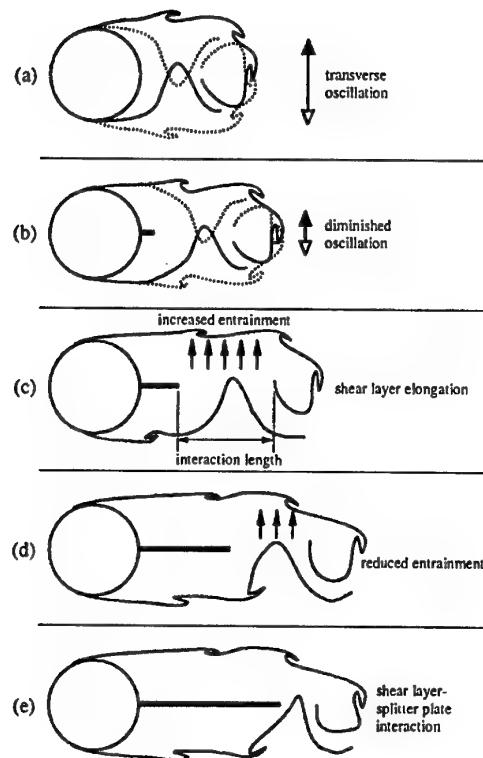


Fig. 7.17 – Summary of the effects of a splitter plate on the formation region behind a circular cylinder. From Anderson and Szewczyk (Anderson and Szewczyk, 1997), by permission.

1994; Cimbala and Leon, 1996), have investigated splitter plates detached from the cylinder and splitter plates that are allowed to freely rotate. The details of these studies are not discussed here, but the changes to the flow characteristics were found to be similar to the attached splitter plate.

Numerical studies of circular cylinders with splitter plates are rather limited. Kwon and Choi (Kwon and Choi, 1995) studied laminar, 2D flow at $Re = 80 - 160$ using a FV method. You et al (You et al., 1998) also used a FV method to investigate laminar, 2D flow—this time at $Re = 100$ and 160. These studies did not show the increase in Strouhal number for very small L/D found by Anderson and Szewczyk (Anderson and Szewczyk, 1997). They do show, however, the significance of the $L/D = 1$ case in terms of Strouhal number and drag coefficient.

In this study, the SFELES solver is used to investigate flow at $Re = 3900$ and splitter plate lengths of $L/D = 0.25, 0.75$, and 1.50 . The behavior of the flowfield in terms of vortex formation location, wake geometry, shedding frequency, and drag coefficient are compared to published experimental studies performed at somewhat higher Reynolds numbers.

7.2.2 Computational Mesh

The computational domain and boundary conditions used for the splitter plate calculations are identical to the no splitter plate case described in the previous section and pictured in Figure 7.1. Due to computational time restraints, a grid refinement study was not feasible. Considering the $L/D = 0$ case of the previous section, it was decided that the element dimensions used in the coarse mesh were sufficient. The only adjustment made was to reduce the spacing along the cylinder and splitter plate walls to $0.02D$ from $0.0275D$. This refinement along the boundary propagates into the shear layer and near-wake regions, providing better resolution. The reader is referred to Section 7.1.2 for information regarding all other element dimensions, including the boundary layer elements. The overall mesh statistics for the three splitter plate cases as well as the $L/D = 0$ coarse grid case of the previous section are given in Table 7.2. The increased resolution along the cylinder and splitter plate results

Table 7.2 – Turbulent circular cylinder splitter plate flow: mesh statistics summary.

L/D	2D Elements	2D Nodes	Transverse Nodes	Overall Nodes
0.00	23,500	12,000	32	380,000
0.25	33,300	16,900	32	540,000
0.75	37,200	18,800	32	602,000
1.50	43,200	21,800	32	699,000

in a substantial increase in overall nodes, especially as the splitter plate length is increased.

Figure 7.18 shows the computational mesh near the cylinder for the $L/D = 0.25$, 0.75 , and 1.50 cases. Here the geometry of the splitter plate as well as the refinement near the walls and in the near wake can be seen. The splitter plate thickness is $\frac{1}{32}D$, consistent with the experimental setup of Anderson and Szewczyk (Anderson and Szewczyk, 1997).

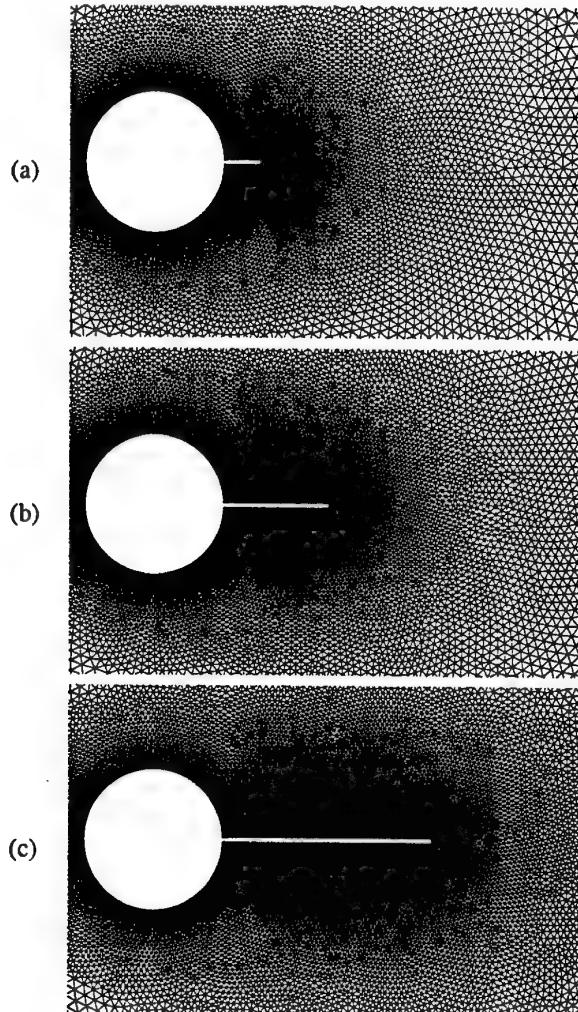


Fig. 7.18 – Turbulent circular cylinder splitter plate flow: in-plane mesh near the cylinder for (a) $L/D = 0.25$, (b) $L/D = 0.75$, and (c) $L/D = 1.50$.

7.2.3 Results

Instantaneous views of the resulting flowfields are shown in Figure 7.19 via surfaces of constant vorticity magnitude. The surfaces are colored according to velocity magnitude, showing the increased quiescence in the recirculation region brought about by even a small

splitter plate. As observed by Apelt et al (Apelt et al., 1973) and Apelt and West (Apelt and West, 1975) for $Re \approx 10^5$, the splitter plate stabilizes the shear layer separation points and produces a narrower wake. For all cases investigated the trend is that as L/D increases, the wake narrows.

The behavior of the near wake region is more closely seen in Figure 7.20. Here, predicted surfaces of vorticity magnitude are compared with flow visualizations from Anderson (Anderson, 1994) at nearly the same Reynolds number ($Re = 3700$). Qualitatively, the length of the formation region is well predicted by SFELES. From the vorticity magnitude isosurfaces it is seen that increasing L/D produces increasing shear layer length and tighter initial roll-up. Referring to the observations of Gerrard (Gerrard, 1966) and Perry et al (Perry et al., 1982) regarding the physical mechanisms of the formation region, the tighter roll-up of the shear layers draws less fluid from the opposite side, resulting in a narrower wake.

Figure 7.21 shows pressure distributions around the cylinder for each splitter plate configuration. These results show little change from the measurements of Anderson and Szewczyk (Anderson and Szewczyk, 1997) at $Re = 46,000$ and Apelt et al (Apelt et al., 1973) at $Re = 20,000$. These profiles show that, though the separation point location does not change significantly, the pressure behind this point varies with L/D . The trend is for increasing base pressure with increasing L/D for L/D up to ≈ 1 , and then decreasing with L/D for $L/D > 1$, with the most significant variations occurring at small L/D .

Table 7.3 shows the variation in pressure drag (C_{DP}), shedding frequency (St), and formation length (L_f) with L/D . The quantities for $L/D = 0$ compare well with experimental measurements as noted in the previous section. The pressure drag on the cylinder decreases with increasing L/D for the $L/D = 0.25$ and $L/D = 0.75$ cases, then increases from $L/D = 0.75$ to $L/D = 1.50$. This is consistent with the experimental observations discussed previously, which indicate $L/D \approx 1$ as the minimum-drag configuration. The shedding frequency increases slightly from $L/D = 0$ to $L/D = 0.25$, then decreases to $L/D = 0.75$ before increasing to $L/D = 1.50$. This trend agrees with the experiments of Anderson and Szewczyk (Anderson and Szewczyk, 1997) and Apelt et al (Apelt et al., 1973) at higher Reynolds numbers, where the shedding frequency was seen to increase for very small splitter plates then decrease to a minimum at approximately $L/D = 1$ and finally increase with L/D for $L/D > 1$. Measurements from Anderson and Szewczyk (Anderson and Szewczyk, 1997) of formation length as a function of L/D show a substantial increase for very small plates, followed by moderate increases in L_f for increasing L/D until around $L/D = 1$, after which L_f remains fairly constant. The SFELES results for L_f in Table 7.3 agree with this experimentally observed trend.

Table 7.3 – Turbulent circular cylinder splitter plate flow: variation of C_{D_P} , St , and L_f with L/D .

L/D	C_{D_P}	$C_{D_P}/C_{D_P(L/D=0)}$	St	L_f
0.00	1.06	100%	0.2179	1.30
0.25	0.962	90.8%	0.2195	1.96
0.75	0.888	83.8%	0.2180	2.51
1.50	0.903	85.3%	0.2222	2.78

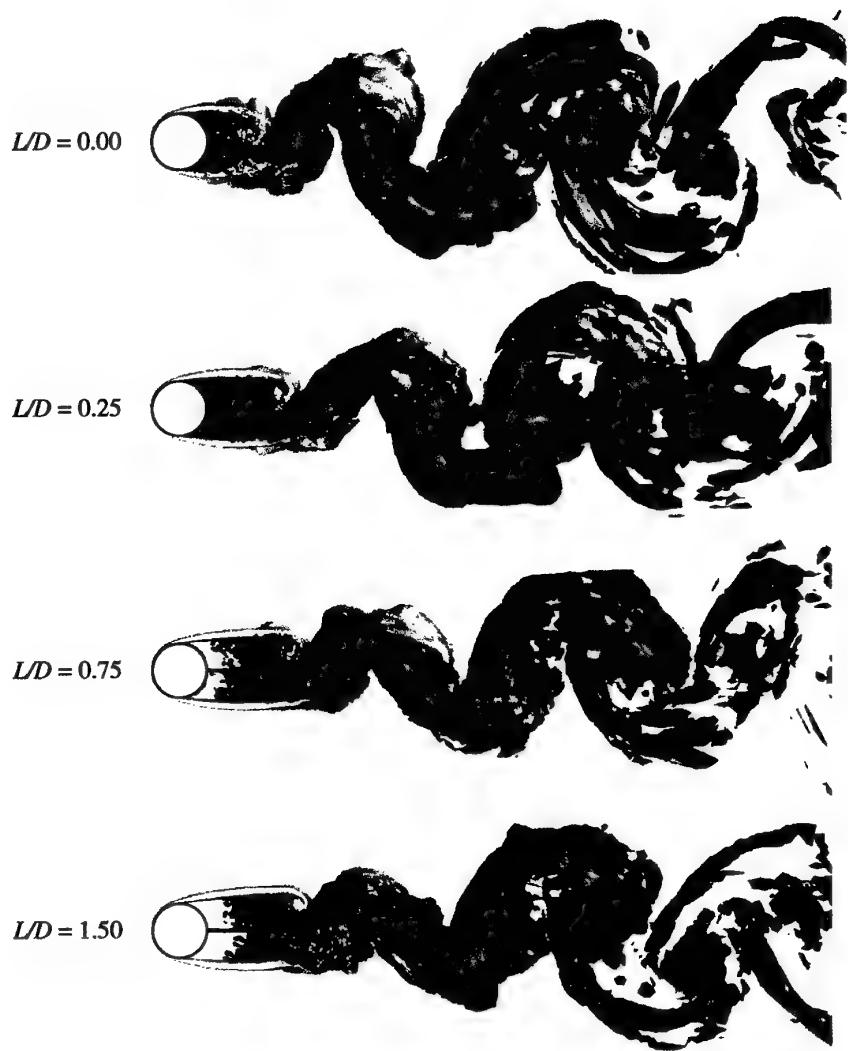


Fig. 7.19 – Turbulent circular cylinder splitter plate flow: instantaneous surfaces of constant vorticity magnitude, $\omega D/U_\infty = 2.5$. Surfaces are colored according to velocity magnitude, with blue representing low values and red representing high values.

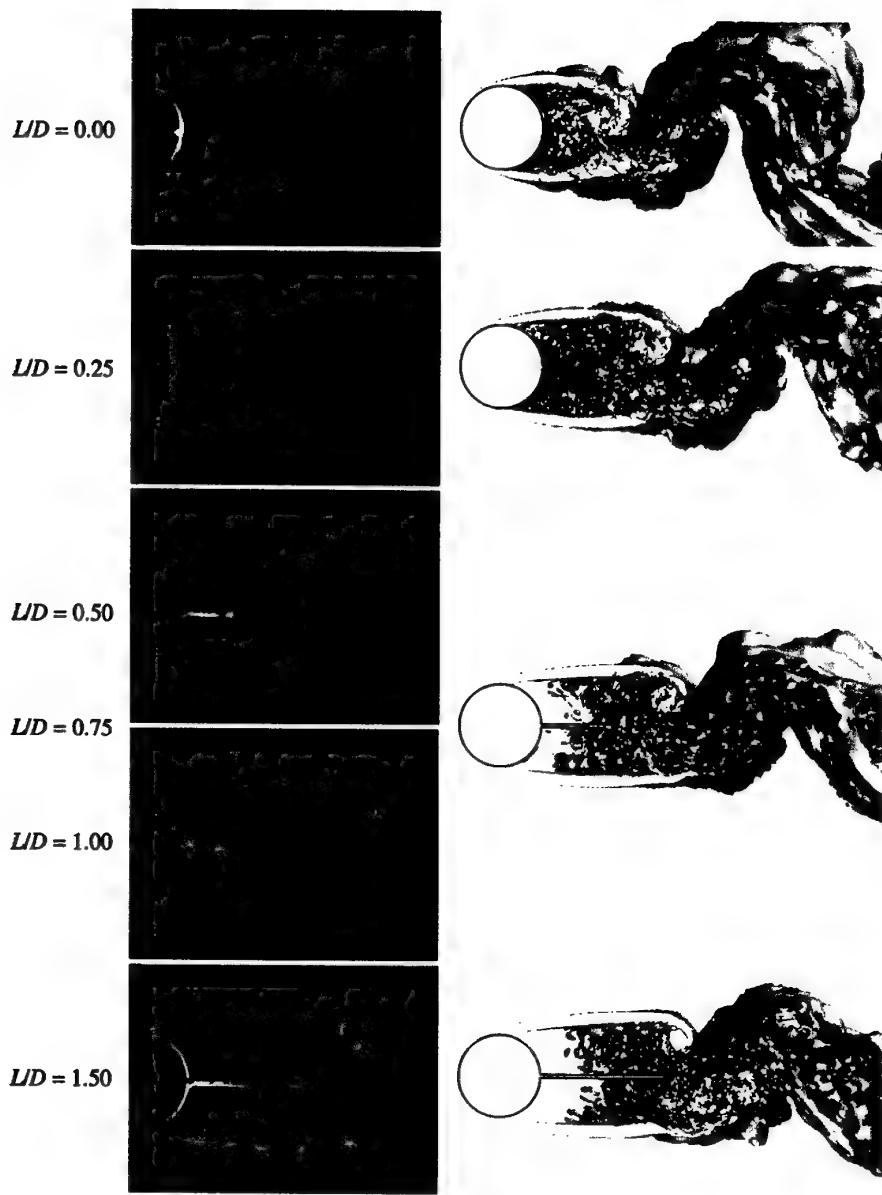


Fig. 7.20 – Turbulent circular cylinder splitter plate flow: effect of splitter plate length on the formation region behind the cylinder. Experimental flow visualizations from Anderson (Anderson, 1994) at $Re = 3700$ (left) compared with predicted surfaces of constant vorticity magnitude at $Re = 3900$ (right).

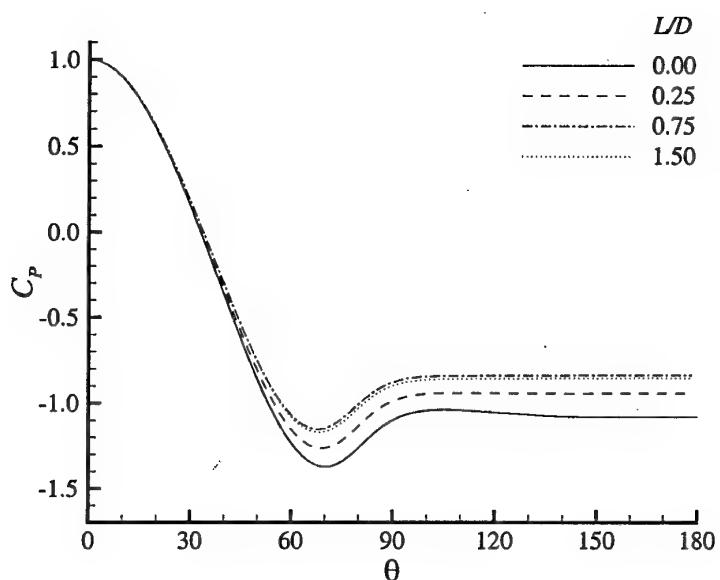


Fig. 7.21 – Turbulent circular cylinder splitter plate flow: pressure distribution around the cylinder for various splitter plate lengths.

8. CONCLUSION

In this concluding chapter the contributions of this work are summarily presented and the future of the work discussed.

8.1 Development of Navier-Stokes Solvers

The ultimate contribution of this work is the development of a parallelized LES solver for use with unstructured meshes. In the course of developing this solver, two additional solvers were developed as intermediate steps to the LES solver.

8.1.1 SFE2D

SFE2D is a 2D, incompressible, laminar, unsteady solver. The spatial discretization is formulated using a SUPG/PSPG FE approach. This formulation has been heavily developed in the last decade and has the advantage of allowing equal-order elements as well as having convective stability without the over-diffusive characteristics of typical upwinding schemes. Linear triangle elements are utilized, resulting in second-order accuracy. As a result of this spatial discretization, very complex geometries can be easily accommodated via unstructured triangle meshes.

The temporal discretization uses a consistent mass matrix and is second-order accurate. An implicit Crank-Nicholson scheme is used for the pressure and diffusion terms, while an explicit second-order Adams-Bashforth scheme is used for the convective terms. Though the explicit treatment of the convective terms introduces a stability limit on the time step size, it affords several advantages when the algorithm is extended to 3D.

Solution of the algebraic system arising at each time step is performed using the SPARSKIT toolkit. This kit is a “black box” GMRES iterative solver package written in Fortran 77 that includes compact matrix storage formats and many preconditioning algorithms. The best performance in terms of robustness and speed in this application was realized using the ILUT preconditioner.

SFE2D was validated against published benchmark lid-driven cavity numerical results at Reynolds numbers between 100 and 5000; as well as published experimental and numerical results for backward-facing step flow at Reynolds numbers between 150 and 1000, and circular cylinder flow at Reynolds numbers between 20 and 140.

8.1.2 SFE3D

SFE3D is a parallelized 3D, incompressible, laminar, unsteady solver. A 2D geometry and periodic flow in the transverse direction are assumed. Keeping in line with this as-

sumption, a FE discretization in the 2D plane is combined with a spectral method in the transverse direction. In order to reduce computational costs, all nonlinear terms are treated using a pseudo-spectral approach where the terms are evaluated in physical space and then transformed into Fourier space. The in-plane and temporal discretizations are identical to the SFE2D solver and are described above. The use of explicit temporal treatment for the nonlinear terms decouples the matrix equations in Fourier space, reducing computational cost and allowing for a novel parallelization scheme where the work is partitioned in Fourier space. The parallelization is implemented using OpenMP, an emerging standard for shared-memory parallelism that allows parallelization with little programming overhead.

SFE3D was validated against published numerical and experimental results for flow past a circular cylinder at Reynolds numbers 195 and 300, where mode-A and mode-B 3D instabilities are present in the wake, respectively.

8.1.3 SFELES

SFELES is a parallelized incompressible LES solver. The classic Smagorinski SGS Reynolds stress model with van Driest damping near the solid walls is used. Because it is a direct extension of SFE3D, SFELES utilizes a second-order SUPG/PSPG formulation in the 2D plane and a spectral formulation in the transverse direction. The temporal accuracy is second-order, with the eddy viscosity terms being treated using an explicit Adams-Bashforth scheme.

An unstructured mesh in the 2D plane allows for LES calculations to be run on complex 2D geometries such as arbitrary bluff bodies, arrays, or complex multi-element airfoils. This capability to feasibly perform LES on unstructured meshes is currently very rare due to the high computational costs associated with both LES and unstructured meshes.

The SFELES solver was validated by comparison with published experimental and numerical studies of flow past a circular cylinder at $Re = 3900$.

8.2 Turbulent Circular Cylinder Flow with a Wake Splitter Plate

The value of unstructured mesh LES capabilities has been illustrated via a brief investigation of turbulent flow past a circular cylinder with an attached wake splitter plate at $Re = 3900$. This geometry configuration (assuming a finite-thickness splitter plate) cannot be well modeled with a single-block structured mesh, but rather needs an unstructured or block-structured mesh. To the author's knowledge, no other numerical results exist to augment the experimental measurements and observations of 3D turbulent flow past cylinders with splitter plates. This work contributes information in terms of the physical mechanisms of vortex formation in the presence of splitter plates as well as shedding frequency and drag characteristics for multiple splitter plate lengths.

8.3 Future Work

This work provides a foundation for much additional research. The topics currently envisioned are discussed below.

8.3.0.0.1 Extension of SFE2D to axisymmetric configurations.

The SFE2D solver can be easily extended to handle axisymmetric configurations, thereby allowing simulation of flows in geometries such as pipes and nozzles.

8.3.0.0.2 Parallelization for distributed memory architectures.

The parallelization scheme developed in this work is well suited for implementation on shared-memory computers only. The recent affordability of distributed memory systems such as PC clusters makes this type of parallelism attractive despite the relative programming difficulty. If the work is partitioned in physical space rather than Fourier space, parallel implementation on a distributed memory system should be rather straight-forward.

8.3.0.0.3 Implementation of wall functions and additional boundary conditions.

In its current form, SFELES contains no wall functions and limited boundary conditions. The addition of wall functions will allow higher Reynolds number simulations to be performed without requiring exorbitant numbers of nodes in the near-wall region. Boundary conditions such as convective outflow and far-field inflow/outflow will also enlarge the class of problems that can be solved with this solver.

8.3.0.0.4 Development of feature detection algorithms.

The ability to identify, isolate, and study coherent structures has become an important part of turbulence research. The SFELES solver provides an excellent “numerical laboratory” for research in feature detection algorithms.

8.3.0.0.5 Investigation of block preconditioners.

Block preconditioners have the potential of cutting the matrix solution time by as much as an order of magnitude. The “black box” solvers investigated in this work (Aztec (Tuminaro et al., 1999) and BILUT (Saad, 2001)) provided no performance improvement—in fact, the matrix solution time was increased by as much as a factor of two. Further investigation of block preconditioners with SFELES may prove very fruitful.

8.3.0.0.6 Implementation of additional SGS Reynolds stress models.

This work has focused on the development of the FE/spectral algorithm rather than on achieving the most accurate LES solutions. The Smagorinski model implemented in SFELES is a very basic SGS model, and has a number of shortcomings (discussed in Section 6.3). SGS model development is an area of intense research, and SFELES can provide an excellent test bed for application of these models to complex geometries.

8.3.0.0.7 Application to airfoil problems.

High-lift aerodynamics has received much attention in the past decade, both in experimental studies and numerical predictions. The difficulties in obtaining acceptable numerical solutions are twofold. First, the multi-element geometry is difficult to mesh properly, particularly when using structured grids. Secondly, the complex separation and reattachment of the flow on multiple bodies is problematic for RANS turbulence models. Once sufficient progress has been made in terms of computational speed, boundary conditions, and SGS models, SFELES will be well-suited for high-lift airfoil performance prediction.

8.3.0.0.8 Extension of SFELES to cylindrical coordinates.

The FE/spectral discretization used in SFELES can easily be adapted to cylindrical coordinates, where a FE discretization is used in the x - r plane and a spectral method used in the θ direction. This would expand the class of problems that can be solved to include turbulent flow in pipes and nozzles, including swirling flows such as vortex breakdown.

Bibliography

- (1998). GAMBIT Modeling Guide. Fluent Incorporated, Lebanon, NH.
- (1998). OpenMP C and C++ Application Program Interface Version 1.0. OpenMP Architecture Review Board. Accessible online:
<http://www.openmp.org/specs/mp-documents/cspec10.ps>.
- (2000). OpenMP Fortran Application Program Interface Version 2.0. OpenMP Architecture Review Board. Accessible online:
<http://www.openmp.org/specs/mp-documents/fspec20.ps>.
- Adve, S. V., Pai, V. S., and Ranganathan, P. (1999). Recent advances in memory consistency models for hardware shared memory systems. In Proceedings of the IEEE, volume 3, pages 445–455.
- Anderson, E. A. (1994). Effects of taper and splitter plates on the near wake characteristics of a circular cylinder in uniform and shear flow. PhD thesis, University of Notre Dame.
- Anderson, E. A. and Szewczyk, A. A. (1997). Effects of a splitter plate on the near wake of a circular cylinder in 2 and 3-dimensional flow configurations. Experiments in Fluids, 23:161–174.
- Apelt, C. J. and West, G. S. (1975). The effects of wake splitter plates on bluff-body flow in the range $10^4 < R < 5 \times 10^4$. Part 2. Journal of Fluid Mechanics, 71:145–160.
- Apelt, C. J., West, G. S., and Szewczyk, A. A. (1973). The effects of wake splitter plates on the flow past a circular cylinder in the range $10^4 < R < 5 \times 10^4$. Journal of Fluid Mechanics, 61:187–198.
- Armaly, B. F., Durst, F., Pereira, J. C. F., and Schönung, B. (1983). Experimental and theoretical investigation of backward-facing step flow. Journal of Fluid Mechanics, 127:473–496.
- Ayotte, K. W. and Taylor, P. A. (1995). A mixed spectral finite-difference 3D model of neutral planetary boundary-layer flow over topography. Journal of the Atmospheric Sciences, 52(20):3523–3537.
- Ayotte, K. W., Xu, D., and Taylor, P. A. (1994). The impact of turbulence closure schemes on predictions of the mixed spectral finite-difference model of flow over topography. Boundary-Layer Meteorology, 68:1–33.
- Babuska, I. (1971). Error bounds for finite element methods. Numerische Mathematik, 16:322–333.

- Babuska, I. (1973). The finite element method with Lagrangian multipliers. Numerische Mathematik, 20:179–192.
- Babuska, I. and Aziz, A. K. (1976). On the angle condition in the finite element method. SIAM Journal on Numerical Analysis, 13(2):214–226.
- Bardina, J., Ferziger, J. H., and Reynolds, W. C. (1980). Improved subgrid model for large-eddy simulation. Technical Report 80-1357, AIAA.
- Batchelor, G. K., Canuto, C., and Chasnov, J. (1992). Homogeneous buoyancy-generated turbulence. Journal of Fluid Mechanics, 235:349–378.
- Beaudan, P. and Moin, P. (1994). Numerical experiments on the flow past a circular cylinder of sub-critical Reynolds number. Technical Report TF-62, Department of Mechanical Engineering, Stanford University.
- Beljaars, A. C. M., Walmsley, J. L., and Taylor, P. A. (1987). A mixed spectral finite difference model of neutrally stratified boundary-layer over roughness changes and topography. Boundary-Layer Meteorology, 38:273–303.
- Berger, E. and Wille, R. (1972). Periodic flow phenomena. Annual Review of Fluid Mechanics, 4:313–340.
- Beudan, P. and Moin, P. (1994). Numerical experiments on the flow past a circular cylinder at sub-critical Reynolds number. Technical Report TF-62, Department of Mechanical Engineering, Stanford University.
- Biringen, S. (1985). Active control of transition by periodic suction-blowing. Physics of Fluids, 27:1345–1347.
- Brezzi, F. (1974). On the existence, uniqueness and approximation of saddle-point problems arising from Lagrangian multipliers. RAIRO, 8:129–151.
- Brooks, A. N. and Hughes, T. J. R. (1982). Streamline-Upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations. Computer Methods in Applied Mechanics and Engineering, 32(1-3):199–259.
- Canuto, C., Hussaini, M. Y., Quarteroni, A., and Zang, T. A. (1988). Spectral Methods in Fluid Dynamics. Springer-Verlag, Berlin.
- Cardell, G. S. (1993). Flow past a circular cylinder with a permeable splitter plate. PhD thesis, California Institute of Technology.
- Chandra, R., Dagum, L., Kohr, D., Maydan, D., McDonald, J., and Menon, R. (2001). Parallel Programming in OpenMP. Morgan Kaufmann, San Francisco.

- Christie, I., Griffiths, D. F., Mitchel, A. R., and Zienkiewicz, O. C. (1976). Finite element methods for second order differential equations with significant first derivatives. International Journal for Numerical Methods in Engineering, 10(6):1389–1396.
- Chyu, C. K. and Rockwell, D. (1996). Near-wake structure of a noscillating cylinder: effect of controlled shear-layer vortices. Journal of Fluid Mechanics, 322:21–49.
- Cimbala, J. M. and Leon, J. (1996). Drag of freely rotatable cylinder/splitter-plate body at subcritical Reynolds number. AIAA Journal, 34:2446–2448.
- Clark, R. A., Ferziger, J. H., and Reynolds, W. C. (1979). Evaluation of subgrid-scale models using an accurately simulated turbulent flow. Journal of Fluid Mechanics, 91:1–16.
- Deardorff, J. W. (1970). A numerical study of three-dimensional turbulent channel flow at large Reynolds numbers. Journal of Fluid Mechanics, 41:453–480.
- DeMulder, T. (1997). Stabilized Finite Element Methods for Turbulent Incompressible Single-Phase and Dispersed Two-Phase Flows. PhD thesis, von Karman Institute for Fluid Dynamics.
- DeMulder, T., Deconinck, H., and Buchlin, J. M. (1994). FEGAS: a finite element solver for 2d viscous incompressible gas flows. Technical Report TN 184, von Karman Institute for Fluid Dynamics, Belgium.
- Dick, E. (1993a). Introduction to finite element techniques in CFD. In VKI Lecture Series: Introduction to Computational Fluid Dynamics. von Karman Institute, Belgium.
- Dick, E. (1993b). Introduction to finite volume techniques in CFD. In VKI Lecture Series: Introduction to Computational Fluid Dynamics. von Karman Institute, Belgium.
- Eidsen, T. M., Hussaini, M. Y., and Zang, T. A. (1986). Simulation of the turbulent Rayleigh-Benard problem using a spectral/finite difference technique. Technical Report CR178027, NASA, Hampton, Va.: NASA Langley Research Center.
- Eliassen, E., Machenauer, E., and Rasmussen, E. (1970). On a numerical method of integration of the hydrodynamical equations with a spectral representation of the horizontal fields. Technical Report 2, Department of Meteorology, Copenhagen University, Denmark.
- Emmons, H. W. (1970). Critique of numerical modeling of fluid-mechanics phenomena. Annual Review of Fluid Mechanics, 2:15–36.
- Ferziger, J. H. and Peric, M. (1999). Computational methods for fluid dynamics. Springer-Verlag, Berlin, 2 edition.
- Fletcher, C. A. J., Glowinski, R., and Holt, M. (1991). Computational Techniques for Fluid Dynamics: Fundamental and General Techniques, volume 1. Springer, New York.

- Franca, L. P. and Frey, S. L. (1992). Stabilized finite element methods: II. The incompressible Navier-Stokes equations. Computer Methods in Applied Mechanics and Engineering, 99:209–233.
- Freitas, C. J. (1995). Perspective: selected benchmarks from commercial CFD codes. Journal of Fluids Engineering, 117:208–218.
- Geist, A., Beguelin, A., Dongarra, J., Jiang, W., Manchek, R., and Sunderam, V. (1994). PVM: Parallel Virtual Machine: A Users' Guide and Tutorial for Networked Parallel Computing. MIT Press, Cambridge.
- Germano, M., Piomelli, U., Moin, P., and Cabot, W. (1991). A dynamic subgrid-scale eddy-viscosity model. Physics of Fluids, 3(7):1760–1765.
- Gerrard, J. H. (1966). The mechanics of the formation region of vortices behind bluff bodies. Journal of Fluid Mechanics, 25:401–413.
- Ghia, U., Ghia, K. N., and Shin, C. T. (1988). High-Re solutions for incompressible flow using the Navier-Stokes equations and a multigrid method. Journal of Computational Physics, 48:387–411.
- Gresho, P. M. and Lee, R. L. (1979). Don't suppress the wiggles—they're telling you something. In Hughes, T. J. R., editor, Finite Element Methods for Convection Dominated Flows, volume 34, pages 37–61. ASME AMD, New York.
- Gresho, P. M. and Sani, R. L. (2000). Incompressible Flow and the Finite Element Method: Advection-Diffusion and Isothermal Laminar Flow. Wiley, New York.
- Hasan, M. A. Z. and Budair, M. O. (1994). Role of splitter plates in modifying cylinder wake flows. AIAA Journal, 32:1992–1998.
- Haworth, D. C. and Jansen, K. (2000). Large-eddy simulation on unstructured deforming meshes: towards reciprocating IC engines. Computers and Fluids, 29:493–524.
- Heinrich, J. C., Huyakorn, P. S., Zienkiewicz, O. C., and Mitchell, A. R. (1977). An ‘upwind’ finite element scheme for two-dimensional convective transport equation. International Journal for Numerical Methods in Engineering, 11(1):131–143.
- Henderson, R. D. (1995). Details of the drag curve near the onset of vortex shedding. Physics of Fluids, 7(9):2102–2104.
- Henderson, R. D. (1999). Adaptive spectral element methods for turbulence and transition. In Barth, T. J. and Deconinck, H., editors, High-Order methods for computational physics, volume 9 of Lecture Notes in Computational Science and Engineering. Springer-Verlag, Berlin.

- Hirsch, C. (1988). Numerical Computation of Internal and External Flows, volume 1. Wiley, New York.
- Hood, P. and Taylor, C. (1974). Navier-Stokes equations using mixed interpolation. In Oden, J. T., editor, Finite element methods in flow problems, pages 121–132. University of Alabama Press, Huntsville.
- Hughes, T. J. R. (1978). A simple scheme for developing ‘upwind’ finite elements. International Journal for Numerical Methods in Engineering, 12(9):1359–1365.
- Hughes, T. J. R. and Atkinson, J. (1978). A variational basis for ‘upwind’ finite elements. In IUTAM Symposium on Variational Methods in the Mechanics of Solids, Evanston, Illinois: Northwestern University.
- Hughes, T. J. R., Franca, L. P., and Balestra, M. (1986). A new finite element formulation for computational fluid dynamics: V. Circumventing the Babuska-Brezzi condition: a stable Petrov-Galerkin formulation of the Stokes problem accommodating equal-order interpolations. Computer Methods in Applied Mechanics and Engineering, 59:85–99.
- Hussaini, M. Y. and Zang, T. A. (1987). Spectral methods in fluid dynamics. Annual Review of Fluid Mechanics, 19:339–367.
- Jansen, K. (1999). A stabilized finite element method for computing turbulence. Computer Methods in Applied Mechanics and Engineering, 174:299–317.
- Kolmogorov, A. N. (1941). The local structure of turbulence in incompressible viscous fluid for very large Reynolds numbers. Doklady Akademii Nauk SSSR, 30:299–303. Russian.
- Kolmogorov, A. N. (1962). A refinement of previous hypotheses concerning the local structure of turbulence in a viscous incompressible fluid at high Reynolds number. Journal of Fluid Mechanics, 13:82–85.
- Kraichnan, R. H. (1976). Eddy viscosity in two and three dimensions. Journal of the Atmospheric Sciences, 33:1521–1536.
- Kravchenko, A. G. and Moin, P. (1997). On the effect of numerical errors in large eddy simulations of turbulent flows. Journal of Computational Physics, 131:310–322.
- Kravchenko, A. G. and Moin, P. (1998). B-spline methods and zonal grids for numerical simulations of turbulent flows. Technical Report TF-73, Stanford University, Stanford, CA.
- Kwon, K. and Choi, H. (1995). Control of laminar vortex shedding behind a circular cylinder using splitter plates. Physics of Fluids, 8:479–486.

- Leonard, A. (1974). Energy cascade in large eddy simulation of turbulent fluid flow. Advances in Geophysics, 18A:237–248.
- Lesieur, M. (1990). Turbulence in Fluids. Kluwer, Dordrecht, 2 edition.
- Lesieur, M. and Metais, O. (1996). New trends in large-eddy simulations of turbulence. Annual Review of Fluid Mechanics, 28:45–82.
- Lilly, D. K. (1967). The representation of small-scale turbulence in numerical simulations. In Proceedings of the IBM Scientific Computing Symposium on Environmental Sciences, pages 195–209.
- Lilly, D. K. (1995). On the application of the eddy viscosity concept in the inertial subrange of turbulence. NCAR Manuscript 123.
- Lourenco, L. M. and Shih, C. (1993). Characteristics of the plane turbulent near wake of a circular cylinder. A particle image velocimetry study. Not published. Results obtained from Kravchenko and Moin (1998).
- Lu, X., Dalton, C., and Zhang, J. (1997). Application of large eddy simulation to an oscillating flow past a circular cylinder. Journal of Fluids Engineering, 119:519–525.
- Mansy, H., Yang, P. M., and Williams, D. R. (1994). Quantitative measurements of three-dimensional structures in the wake of a circular cylinder. Journal of Fluid Mechanics, 270:227–296.
- McMillan, O. J., Ferziger, J. H., and Rogallo, R. S. (1980). Tests of new subgrid-scale models in strained turbulence. Technical Report 80-1339, AIAA.
- Message Passing Interface Forum (1995). MPI: A message-passing interface standard. Accessible online at <ftp://ftp.mcs.anl.gov/pub/mpi/mpi-1.jun95/mpi-report.ps>.
- Metais, O. and Lesieur, M. (1989). Large-eddy simulations of isotropic and stably-stratified turbulence. In Fernholz, H. H. and Fiedler, H. E., editors, Advances in Turbulence, pages 371–376. Springer-Verlag, Berlin, 2 edition.
- Metais, O. and Lesieur, M. (1992). Spectral large-eddy simulations of isotropic and stably-stratified turbulence. Journal of Fluid Mechanics, 239:157–194.
- Mitall, R. (2001). Computation of three-dimensional flows past circular cylinder of low aspect ratio. Physics of Fluids, 13(1):177–191.
- Mitall, R. and Balachandar, S. (1995). Effect of three-dimensionality on the lift and drag of nominally two-dimensional cylinders. Physics of Fluids, 7(8):1841–1865.
- Mitall, R. and Moin, P. (1997). Suitability of upwind-biased finite-difference schemes for large-eddy simulation of turbulent flows. AIAA Journal, 35(8):1415–1417.

Mittal, R. and Balachandar, S. (1997). On the inclusion of three-dimensional effects in simulations of two-dimensional bluff body wake flows. In Proceedings of the ASME Fluids Engineering Division Summer Meeting, Vancouver, British Columbia, Canada.

Miura, S., Kakuda, K., and Tosaka, N. (1999). Large eddy simulation of flow around a rectangular cylinder using the finite element method. International Journal for Computational Fluid Dynamics, 12:291–299.

Mizukami, A. (1985). An implementation of the Streamline-Upwind/Petrov-Galerkin method for linear triangular elements. Computer Methods in Applied Mechanics and Engineering, 49(3):357–364.

Moin, P. and Kim, J. (1982). Numerical investigation of turbulent channel flow. Journal of Fluid Mechanics, 118:341–377.

Moin, P., Squires, K., Cabot, W., and Lee, S. (1991). A dynamic subgrid-scale model for compressible turbulence and scalar transport. Physics of Fluids A, 3:2746–2757.

Morkovin, M. V. (1964). Flow around a circular cylinder—a kaleidoscope of challenging fluid phenomenon. In Proceedings of the ASME Symposium on Fully Separated Flows, pages 102–119, Philadelphia, PA.

Müller, J. D., Roe, P. L., and Deconinck, H. (1993). A frontal approach for internal node generation in Delaunay triangulations. International Journal for Numerical Methods in Fluids, 17(3):241–255.

Norberg, C. (1993). Pressure forces on a circular cylinder in cross flow. In Eckelmann, H., Graham, J. M. R., Heurre, P., and Monkewitz, P. A., editors, Proceedings of IUTAM Symposium on Bluff-Body Wakes, Dynamics and Instabilities, pages 275–278. Springer-Verlag, Berlin.

Ong, L. and Wallace, J. (1996). The velocity field of the turbulent very near wake of a circular cylinder. Experiments in Fluids, 20:441–453.

Orzag, S. A. (1969). Numerical methods for the simulation of turbulence. Physics of Fluids, 12 (Suppliment II):250–257.

Orzag, S. A. (1971a). Numerical simulation of incompressible flows within simple boundaries: I. Galerkin (spectral) representations. Studies in Applied Mathematics, 50:293–327.

Orzag, S. A. (1971b). On the elimination of aliasing in finite-difference schemes by filtering high-wavenumber components. Journal of the Atmospheric Sciences, 28:1074.

Pacheco, P. (1996). Parallel Programming with MPI. Morgan Kaufmann, San Francisco.

- Patera, A. T. (1984). A spectral element method for fluid dynamics: laminar flow in a channel expansion. Journal of Computational Physics, 54:468–488.
- Patterson, G. S. and Orzag, S. A. (1971). Spectral calculations of isotropic turbulence: efficient removal of aliasing interactions. Physics of Fluids, 14:2538–2541.
- Perry, A. E., Chong, M. S., and Lim, T. T. (1982). The vortex-shedding process behind two-dimensional bluff bodies. Journal of Fluid Mechanics, 116:77–90.
- Piomelli, U. (1993). High Reynolds number calculations using the dynamic subgrid-scale stress model. Physics of Fluids A, 5:1484–1490.
- Piomelli, U. (1994). Large-eddy simulation of turbulent flows. Technical Report 767, TAM.
- Piomelli, U., Zang, T. A., Speziale, C. G., and Hussiani, M. Y. (1990). On the large-eddy simulation of transitional wall-bounded flows. Physics of Fluids A, 2:257–265.
- Prasad, A. and Williamson, C. H. K. (1997). The instability of the shear layer separating from a bluff body. Journal of Fluid Mechanics, 333:375–402.
- Press, W. H., Teukolsky, S. A., Vetterling, W. T., and Flannery, B. P. (1992). Numerical Recipes in Fortran 77. Cambridge University Press, Cambridge.
- Reynolds, O. (1895). On the dynamical theory of incompressible viscous fluids and the determination of the criterion. Philosophical Transactions of the Royal Society of London (Series A), 186:123–164.
- Roshko, A. (1954). On the drag and shedding frequency of two-dimensional bluff bodies. Technical Report 3169, NACA.
- Saad, Y. (1981). Krylov subspace methods for solving large unsymmetric linear systems. Mathematics of Computation, 37:105–126.
- Saad, Y. (1994a). ILUT: a dual threshold incomplete ILU preconditioner. Numerical Linear Algebra with Applications, 1:387–402.
- Saad, Y. (1994b). SPARSEKIT: a basic tool kit for sparse matrix computations. Department of Computer Science and Engineering, University of Minnesota.
- Saad, Y. (2001). Private communication.
- Saad, Y. and Schultz, M. H. (1986). GMRES: a generalized minimum residual algorithm for solving nonsymmetric linear systems. SIAM Journal on Scientific and Statistical Computing, 7(3):856–869.
- Schlichting, H. (1979). Boundary Layer Theory. McGraw-Hill, New York.

- Silberman, I. (1954). Planetary waves in the atmosphere. Journal of Meteorology, 11:27–34.
- Smagorinski, J. (1963). General circulation experiments with the primitive equations part I: the basic experiment. Monthly Weather Review, 91:99–164.
- Son, J. and Hanratty, T. J. (1969). Velocity gradients at the wall for flow around a cylinder at Reynolds numbers 5×10^3 to 10^5 . Journal of Fluid Mechanics, 35:353–368.
- Squires, K. D. and Piomelli, U. (1994). Dynamic modeling of rotating turbulence. In Durst, F., Kasagi, N., Launder, B. E., Schmidt, F. W., and Whitelaw, J. H., editors, Turbulent Shear Flows, volume 9, pages 71–84. Springer-Verlag, Heidelberg.
- Still, C. H., Langer, S. H., and Alley, W. E. (1998). Shared-memory programming with OpenMP. Computers in Physics, 12(6):577–584.
- Taylor, C. and Hood, P. (1973). A numerical solution of the Navier-Stokes equations using FEM technique. Computers and Fluids, 1:73–100.
- Tezduyar, T. E. and Hughes, T. J. R. (1983). Finite element formulations for convection dominated flows with particular emphasis on the compressible Euler equations. In Proceedings of the AIAA 21st Aerospace Sciences Meeting, Reno, Nevada.
- Tezduyar, T. E., Mittal, S., Ray, S. E., and Shih, R. (1992). Incompressible flow computations with stabilized bilinear and linear equal-order interpolation velocity-pressure elements. Computer Methods in Applied Mechanics and Engineering, 95(2):221–242.
- Tezduyar, T. E., Mittal, S., and Shih, R. (1991). Time-accurate incompressible flow computations with quadrilateral velocity-pressure elements. Computer Methods in Applied Mechanics and Engineering, 87:363–384.
- Thompson, M., Hourigan, K., and Sheridan, J. (1994). Three-dimensional instabilities in the cylinder wake. In International Colloquium on Jets, Wakes, and Shear Layers, Melbourne, Australia.
- Throop, J. (1999). OpenMP: shared-memory parallelism from the ashes. Computer, 32(5):108–109.
- Tritton, D. (1959). Experiments on the flow past a circular cylinder at low Reynolds numbers. Journal of Fluid Mechanics, 6:547–567.
- Tuminary, R. S., Heroux, M., Hutchinson, S. A., and Shadid, J. N. (1999). Official aztec user's guide version 2.1. Sandia National Laboratories. SAND 99-8801J.
- Unal, M. F. and Rockwell, D. (1987). On vortex formation from a cylinder. Part 2. Control by splitter-plate interference. Journal of Fluid Mechanics, 190:513–529.

- VanDyke, M. (1982). An Album of Fluid Motion. Parabolic Press, Stanford, CA.
- Versteeg, H. K. and Malalasekera, W. (1995). An introduction to computational fluid dynamics: the finite volume method. Wiley, New York.
- Weiselsberger, C. (1922). New data on the laws of fluid resistance. Technical Report TN-84, NACA.
- Williamson, C. H. K. (1988). Defining a universal and continuous Strouhal-Reynolds number relationship for the laminar vortex shedding of a circular cylinder. Physics of Fluids, 31(10):2742–2744.
- Williamson, C. H. K. (1996a). Three dimensional wake transition. Journal of Fluid Mechanics, 328:345–407.
- Williamson, C. H. K. (1996b). Vortex dynamics in the cylinder wake. Annual Review of Fluid Mechanics, 28:477–539.
- Williamson, C. H. K. and Roshko, A. (1990). Measurements of base pressure in the wake of a cylinder at low Reynolds numbers. Zeitschrift fur Flugwiss, 14:38–52.
- Williamson, C. H. K., Wu, J., and Sheridan, J. (1995). Scaling of streamwise vortices in wakes. Physics of Fluids, 7(10):2307–2309.
- Wray, A. and Hussaini, M. Y. (1984). Numerical experiments in boundary-layer stability. Proceedings of the Royal Society of London Series A, 392:373–389.
- You, D., Choi, H., Choi, M., and Kang, S. (1998). Control of flow-induced noise behind a circular cylinder using splitter plates. AIAA Journal, 36:1961–1967.
- Yuan, H. and Prosperetti, A. (1994). On the in-line motion of two spherical bubbles in a viscous fluid. Journal of Fluid Mechanics, 278:325–349.
- Zang, T. A. (1991). On the rotation and skew-symmetric forms for incompressible flow simulations. Applied Numerical Mathematics, 7:27–40.
- Zang, T. A. and Piomelli, U. (1993). Large eddy simulation of transitional flow. In Galperin, B. and Orzag, S. A., editors, Large Eddy Simulation of Complex Engineering and Geophysical Flows, pages 209–230. Cambridge University Press, Cambridge.
- Zienkiewicz, O. C. (1975). Why finite elements? In Gallagher, R. H., Oden, J. T., Taylor, C., and Zienkiewicz, O. C., editors, Finite Elements in Fluids, volume 1, pages 1–23. Wiley, London.

APPENDIX

A. COMPRESSED SPARSE ROW (CSR) MATRIX FORMAT

The CSR format is the basic matrix storage format used in the SPARSKIT package, which is used to solve linear systems generated at each time step. In order to avoid extra computational cost in conversion between formats, it is also the native storage format utilized inside the SFELES solver.

The CSR data structure consists of three arrays: the double precision array **A** and two integer arrays **IA** and **JA**. Considering the general matrix

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1N} \\ a_{21} & a_{22} & \cdots & a_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ a_{M1} & \cdots & \cdots & a_{MN} \end{bmatrix}$$

with NNZ non-zero a_{ij} values. The matrix is stored as

A[1...NNZ] where

A[k] = Real value of the k th non-zero a_{ij} ,
stored row by row, from row 1 to row M

JA[1...NNZ] where

JA[k] = Column index corresponding to the value
in **A[k]**

IA[1...N + 1] where

IA[n] = Pointer to the beginning of row n in the
A and **JA** arrays

As an example, consider the 3×3 array with five non-zero values

$$\begin{bmatrix} 1.0 & 2.0 \\ 3.0 & 4.0 \\ 5.0 \end{bmatrix}.$$

The **A** array contains the nonzero values ordered row by row

$$A = [1.0 | 2.0 | 3.0 | 4.0 | 5.0].$$

The **JA** array contains the column indices corresponding to the **A** values

$$JA = [1 | 2 | 2 | 3 | 1].$$

Finally, the **IA** array points to the beginning indices of the rows of the matrix in **A** and **JA**

$$IA = [1 | 3 | 5 | 6].$$

Though this matrix storage format may be somewhat complex in terms of implementation, it provides profound savings in terms of memory requirements for sparse matrices.

B. INPUT DATA FILE FORMAT (2D)

The DPlot (.dpl) file format, a rather compact and versatile ASCII finite-element format, is used as the input/output format for the 2D code. The format is described below via the .dpl file listing of a simple example and the corresponding grid shown in Figure B.1.

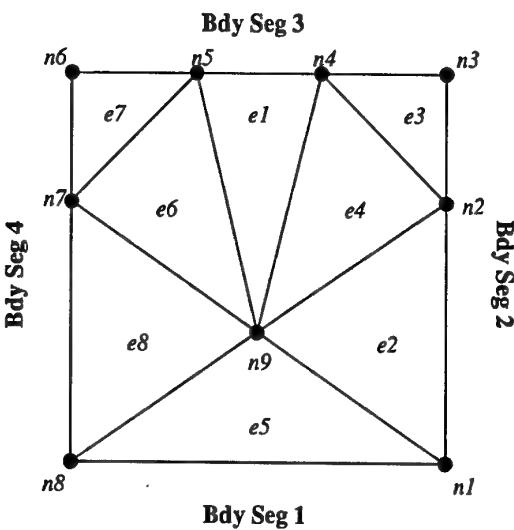


Fig. B.1 – Mesh corresponding to the sample DPlot data file, where e denotes an element and n denotes a node.

unstructured grid data

```

8 9 8
3 9 4 5 13 6 4 1
3 9 1 2 10 4 5 2
3 4 2 3 11 12 4 3
3 9 2 4 3 1 2 4
3 9 8 1 9 2 8 5
3 9 5 7 7 8 1 6
3 7 5 6 14 15 6 7
3 9 7 8 16 5 6 8
9

```

(1)

(2)

(3)

```
1. 1. 1. 1. 0. 0.
```

(4)

```
0.3000e+01 0.0000e+00 1. 1. 1. 1. 1
```

(5)

```
0.3000e+01 0.2000e+01 1. 1. 1. 1. 2
```

(6)

```
0.3000e+01 0.3000e+01 1. 1. 1. 1. 3
```

```
0.2000e+01 0.3000e+01 1. 1. 1. 1. 4
```

0.1000e+01 0.3000e+01	1.	1.	1.	1.	5	
0.0000e+00 0.3000e+01	1.	1.	1.	1.	6	
0.0000e+00 0.2000e+01	1.	1.	1.	1.	7	
0.0000e+00 0.0000e+00	1.	1.	1.	1.	8	
0.1268e+01 0.1000e+01	1.	1.	1.	1.	9	
4 4 1 3 6 8 2 4 3 1						(7)
1 1						(8)
1 8 1 1 5 1						(9)
2 2						
2 1 1 1 2 1						
3 2 1 1 3 2						
3 3						
4 3 1 1 3 1						
5 4 1 1 1 2						
6 5 1 1 7 3						
2 4						
7 6 1 1 7 1						
8 7 1 1 8 2						
0 -1						(10)

An explanation of each section is now given.

1. A text header.
2. The number of elements, the number of nodes, and the number of boundary faces.
3. For each element, the number of nodes, the node numbers at each corner in the counterclockwise sense, the neighboring elements with neighbor #1 being opposite node #1, and a running counter.
4. The number of nodes.
5. Four reference free-stream state quantities in conservation variables and two placeholders.
6. For each node, the x and y coordinates, four state quantities in conservation variables, and a running counter.
7. The number of boundary segments, the number of corners between boundary segments, the node numbers at each corner, and the left-oriented boundary name for each corner, i.e. the name of the boundary oriented with the domain to the left, having this node as its last node.

8. For each segment, the number of boundary faces along the segment and the number of the segment. Each segment must be described such that the mesh is on the left.
9. For each boundary face, the two forming nodes in opposite sense to the description of the segment, the boundary condition type for the first node, the boundary condition type for the second node, the neighboring element, and a running counter.
10. An end-of-file indicator.

C. INPUT DATA FILE FORMAT (3D)

A modified version of the DPlot (.dpl) file format was developed for use as the input/output format for the SFELES solver. The format is described below via the .dpl file listing of a simple example. The 2D mesh is identical to the example mesh of Appendix C, but in this case the flowfield variables are stored at four equi-spaced planes, as shown in Figure C.1.

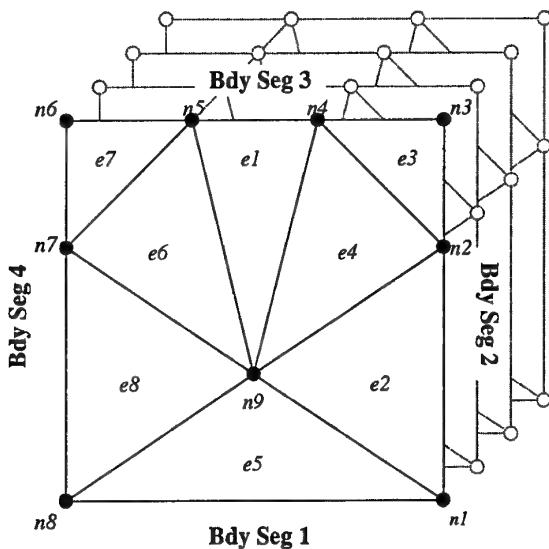


Fig. C.1 – Mesh corresponding to the sample modified DPlot data file, where e denotes an element and n denotes a node.

unstructured grid data	(1)
8 0 4	(2)
3 9 4 5 13 6 4 1	(3)
3 9 1 2 10 4 5 2	
3 4 2 3 11 12 4 3	
3 9 2 4 3 1 2 4	
3 9 8 1 9 2 8 5	
3 9 5 7 7 8 1 6	
3 7 5 6 14 15 6 7	
3 9 7 8 16 5 6 8	
9	(4)
1. 1. 1. 0. 0.	(5)
0.3000e+01 0.0000e+00 1. 1. 1. 1.	

		1.	1.	1.	1.	
		1.	1.	1.	1.	
		1.	1.	1.	1.	
		0.	0.	0.	0.	
		0.	0.	0.	0.	
		0.	0.	0.	0.	
		0.	0.	0.	0.	1
0.3000e+01	0.2000e+01	1.	1.	1.	1.	
		1.	1.	1.	1.	
		1.	1.	1.	1.	
		1.	1.	1.	1.	
		0.	0.	0.	0.	
		0.	0.	0.	0.	
		0.	0.	0.	0.	2
0.3000e+01	0.3000e+01	1.	1.	1.	1.	
		1.	1.	1.	1.	
		1.	1.	1.	1.	
		1.	1.	1.	1.	
		0.	0.	0.	0.	
		0.	0.	0.	0.	
		0.	0.	0.	0.	3
0.2000e+01	0.3000e+01	1.	1.	1.	1.	
		1.	1.	1.	1.	
		1.	1.	1.	1.	
		1.	1.	1.	1.	
		0.	0.	0.	0.	
		0.	0.	0.	0.	
		0.	0.	0.	0.	4
0.1000e+01	0.3000e+01	1.	1.	1.	1.	
		1.	1.	1.	1.	
		1.	1.	1.	1.	
		1.	1.	1.	1.	
		0.	0.	0.	0.	
		0.	0.	0.	0.	
		0.	0.	0.	0.	5
0.0000e+00	0.3000e+01	1.	1.	1.	1.	
		1.	1.	1.	1.	

(7)
(8)
(9)

7 6 1 1 7 1
8 7 1 1 8 2
0 -1

(10)

An explanation of each section is now given.

1. A text header.
2. The number of elements, an unused placeholder, and the number of 2D planes (Fourier modes).
3. For each 2D element, the number of nodes, the node numbers at each corner in the counterclockwise sense, the neighboring elements with neighbor #1 being opposite node #1, and a running counter.
4. The number of nodes.
5. Four reference free-stream state quantities in conservation variables and two placeholders.
6. For each node, the x and y coordinates, four state quantities in conservation variables for each two dimensional plane and time steps n and $n - 1$ (i.e. the four variables first for plane 0 and time step n , then plane 1 and time step $n, \dots, \text{plane } 3 \text{ and time step } n$, plane 0 and time step $n - 1, \dots, \text{plane } 3 \text{ and time step } n - 1$), and a running counter.
7. The number of boundary segments, the number of corners between boundary segments, the node numbers at each corner, and the left-oriented boundary name for each corner, i.e. the name of the boundary oriented with the domain to the left, having this node as its last node.
8. For each segment, the number of boundary faces along the segment and the number of the segment. Each segment must be described such that the mesh is on the left.
9. For each boundary face, the two forming nodes in opposite sense to the description of the segment, the boundary condition type for the first node, the boundary condition type for the second node, the neighboring element, and a running counter.
10. An end-of-file indicator.

D. COMMUTATIVE PROPERTIES OF THE LES FILTER OPERATION

The LES filtering operation given in (6.2) is repeated here:

$$\bar{f}(\vec{x}, t) = \int_{\Omega} f(\vec{r}, t) G(\vec{x} - \vec{r}) d\vec{r}. \quad (\text{D.1})$$

In the derivation of the LES governing equations (Section 6.2), it was stated that the filtering operation commutes with both spatial and temporal derivatives, as is now shown.

First, the temporal derivative,

$$\frac{\partial \bar{f}}{\partial t} \stackrel{?}{=} \frac{\partial \bar{f}}{\partial t}. \quad (\text{D.2})$$

Substituting in the definition of the filter operation (D.1) gives

$$\frac{\partial}{\partial t} \int_{\Omega} G(x_i - r_i) f(r_i, t) dr_i \stackrel{?}{=} \int_{\Omega} G(x_i - r_i) \frac{\partial f(r_i, t)}{\partial t} dr_i. \quad (\text{D.3})$$

Assuming the integral on the LHS is temporally continuous, the derivative can be moved inside the integral:

$$\int_{\Omega} \frac{\partial}{\partial t} [G(x_i - r_i) f(r_i, t)] dr_i \stackrel{?}{=} \int_{\Omega} G(x_i - r_i) \frac{\partial f(r_i, t)}{\partial t} dr_i. \quad (\text{D.4})$$

Finally, because the filter function G is not a function of time we see that

$$\int_{\Omega} G(x_i - r_i) \frac{\partial f(r_i, t)}{\partial t} dr_i = \int_{\Omega} G(x_i - r_i) \frac{\partial f(r_i, t)}{\partial t} dr_i, \quad (\text{D.5})$$

$$(\text{D.6})$$

and the filtering operation does, in fact, commute with temporal derivatives.

Now the commutative property of the spatial derivative is verified:

$$\frac{\partial \bar{f}}{\partial x_i} \stackrel{?}{=} \frac{\partial \bar{f}}{\partial x_i}. \quad (\text{D.7})$$

Again, substituting in the definition of the filter operation (D.1) gives

$$\frac{\partial}{\partial x_i} \int_{\Omega} G(x_i - r_i) f(r_i, t) dr_i \stackrel{?}{=} \int_{\Omega} G(x_i - r_i) \frac{\partial f(r_i, t)}{\partial r_i} dr_i. \quad (\text{D.8})$$

Assuming the integrand on the LHS is spatially continuous, the derivative can be moved inside the integral. The RHS can be integrated by parts to obtain

$$\int_{\Omega} \frac{\partial}{\partial x_i} [G(x_i - r_i) f(r_i, t)] dr_i \stackrel{?}{=} [G(x_i - r_i) f(r_i, t)]_{\Gamma} - \int_{\Omega} f(r_i, t) \frac{\partial G(x_i - r_i)}{\partial r_i} dr_i, \quad (\text{D.9})$$

where the Γ denotes the boundary of the domain. This term is identically zero as long as G is zero on the boundary. Dropping this term and rearranging the LHS gives

$$\int_{\Omega} \frac{\partial G(x_i - r_i)}{\partial x_i} f(r_i, t) dr_i \stackrel{?}{=} - \int_{\Omega} f(r_i, t) \frac{\partial G(x_i - r_i)}{\partial r_i} dr_i. \quad (\text{D.10})$$

Because $G = G(x_i - r_i)$, we have $\partial G / \partial x_i = -\partial G / \partial r_i$. Applying this gives the desired result

$$\int_{\Omega} \frac{\partial G(x_i - r_i)}{\partial x_i} f(r_i, t) dr_i = \int_{\Omega} \frac{\partial G(x_i - r_i)}{\partial x_i} f(r_i, t) dr_i, \quad (\text{D.11})$$

and we see that the filtering operation does, in fact, commute with spatial derivatives.

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY (Leave blank)			2. REPORT DATE		3. REPORT TYPE AND DATES COVERED		
			December 2002		Final Technical Report		
4. TITLE AND SUBTITLE LES investigation of coherent structures in boundary layers and wakes Volume III: Development of a Parallel Unstructured Grid LES Code					5. FUNDING NUMBERS Grant No. N0014-99-1-0834		
6. AUTHOR(S) G. Degrez & D. Snyder							
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Institut von Karman de Dynamique des Fluides (Von Karman Institute for Fluid Dynamics) Chaussee de Waterloo, 72 1640 Rhode-Saint-Genese, BELGIUM					8. Performing Organization Report Number Grant No. : N00014-99-1-0834		
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) Office of Naval Research, Ballston Centre Tower, 800 North Quincy Street Arlington, VA 22217-5660					10. SPONSORING/MONITORING AGENCY REPORT NUMBER EAR9944		
11. SUPPLEMENTARY NOTES This research was sponsored by the Office of Naval Research (ONR Grant Number: N00014-99-1-0834; Contract Report 2003-08). Reproduction in whole or in part is permitted for any purpose of the United States Government. Part of a three volume set -- LES investigation of coherent structures in boundary layers and wakes -- Volume I: Investigation of coherent structure in an attached shear layer (240 pages); Volume II: Wake Around Square Cylinder (164 pages); Volume III: Development of a Parallel Unstructured Grid LES Code (173 pages).							
12a. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution is unlimited.				12b. DISTRIBUTION CODE A			
ABSTRACT (Maximum 200 words) The purpose of the present investigation is to assess the feasibility of simulating and studying coherent structures in turbulent shear layers, making use of Large Eddy Simulations (LES). Volumes I and II were devoted to the investigation of coherent structures in LES solutions obtained using a structured-grid finite-difference code, first in an attached turbulent flow, and then in the wake behind a square cylinder. The structured grid code used in Parts I & II, despite its multi-domain capability, is limited in the domain geometries it can handle. The third part of the work was therefore devoted to the development of an unstructured grid LES code able to compute turbulent flows over arbitrary two-dimensional geometries. A combined finite-element/Fourier spectral space discretization scheme was selected, as it combines optimally the geometrical flexibility provided by the finite element scheme and the computational efficiency resulting from the decomposition in Fourier modes in the out-of-plane direction. Indeed, thanks to this decomposition, and a suitable treatment of the nonlinear convective terms, the 3D flow problem is transformed into a series of 2D problems in Fourier space, completely decoupled within each time step. In addition, the decoupling allows for an easy parallelization by partitioning the work in Fourier space rather than physical space. Also, the stabilized finite element technique selected for the in-plane discretization provides an accuracy superior to its finite volume counterpart on the same unstructured grid and for the same discretization stencil. The development of the code was broken down in the following steps: 1. development of a two-dimensional unsteady laminar flow solver, and validation; 2. development of a three-dimensional unsteady laminar flow solver, and validation,- 3. development of the three-dimensional LES code, validation, and application to an original flow problem. Although originally planned, the extension of the coherent structures detection algorithm developed in Part I for the structured grid LES solver could not be carried out for lack of time, as the development and testing of the solver took more time than anticipated. The computational results obtained with the present solver were found in excellent agreement with existing experimental and computational data for the test case of the flow over a circular cylinder. The original computation of the flow over a circular cylinder with splitter plate was also found to be in excellent agreement with experimental data. These encouraging results suggest that the developed code has a great potential for further investigation of turbulent flows over general two-dimensional geometries. Nevertheless, improvements are possible and needed. Suggestions for such improvements, and further investigations are provided.							
14. SUBJECT TERMS ONR, Belgium, Large-Eddy simulations, Fluid dynamics, Finite-element method, Spectral method, 2D solver, 3D Laminar solver, LES solver, Spatial discretization, Temporal discretization, Navier-Stokes solver					15. NUMBER OF PAGES 16. PRICE CODE		
17. SECURITY CLASSIFICATION OF REPORT UNCLASSIFIED		18. SECURITY CLASSIFICATION OF THIS PAGE UNCLASSIFIED		19. SECURITY CLASSIFICATION OF ABSTRACT UNCLASSIFIED		20. LIMITATION OF ABSTRACT UL	

2980nrc0f603c



von KARMAN INSTITUTE FOR FLUID DYNAMICS INPA
INSTITUT von KARMAN DE DYNAMIQUE DES FLUIDES AISBL
von KARMAN INSTITUUT VOOR STROMINGSDYNAMICA IVZW

REF.

Dr. Patrick Purtell
Office Naval Research
Ballston Center Tower One,
800 North Quincy Street
Arlington VA 22217-5660
U.S.A.

Administrative Office
Office of Naval Research Regional Office Boston
495 Summer Street Room 627
Boston MA 02210-2109
U.S.A

Defense Technical Information Center
8725 John J. Kingman Road Ste 0994
Fort Belvoir VA 22060-6218
U.S.A.

Office of Naval Research
ATTN: ONR 363
Ballston Center Tower One,
800 North Quincy Street
Arlington VA 22217-5660
U.S.A.

Patent Office
Office of Naval Research
ATTN: ONR 00CC
Ballston Center Tower One,
800 North Quincy Street
Arlington VA 22217-5660
U.S.A.